UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU D'ENGINYIERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Towards routine discovery from egocentric photostreams

Autor: Alexander Bevzenko

Directora: Estefanía Talavera M.
Realitzat a: Dept. de Matemàtiques i Informàtica
Barcelona, February 1, 2019

Resume

When I was thinking what topic to choose and what work would it be. On the one hand I wanted it to be from the area that is particularly interesting and I can use my knowledge. On the other hand I wanted to study something new and exciting. This work exceeded my expectations.

In this work I apply all my skills developed during my education in University of Barcelona to work on real scientific case. I do my research reading many papers on topic, develop methods that suit my case, brainstorm new ideas, spend hours coding and debugging them, spend days waiting calculations to finish and finally - obtain results I'm satisfied with.

This work is about turning 106 days and 70GB of egocentric photostreams into a simple predictions if a particular day is a routine or no.

Gratitudes

I want to thank my family for supporting me all that time, my friends for enduring me,
my tutor. Estefanía Talavera for helping me to surpass difficulties and always being ready to answer my questions,
all professors and students of University of Barcelona for aiding me on my way to knowledge,
my cat for sitting on my belly.

Contents

1 Introduction

People's daily life can be exciting, it can have sudden events or it can be a routine. In this project I aim to detect latter. But what exactly is a routine?

According to Cambridge Dictionary routine is: a usual or fixed way of doing things[1]. So by definition - routine days must be similar to each other and non routines must be different. It doesn't mean that routine days must be 'boring', or non-routines are great days. Also comparing different people days has no sense. What is clearly a routine for one, may be unusual day for another.

The main idea is that by capturing a person's daily life we will be able to extract patterns and after analysis tell what days are "usual" ones and what days are different.

It has many applications even beyond routine detection: like health related researches. And it is actively growing area of interest - there are increasing number of works on topic, like [2] and [3] and [4].

1.1 What is lifelogging?

Lifelogging is the act of capturing moments of your life. With today's technology recording all parts of your daily life is very easy. With all kinds of wearable electronics you can log anything from what you see up to your pulse.

The act of recording moments of a person's life became popular in last years. For instance, many youtubers rely on it as their main source of content - for example travel related ones: they wear an action camera, usually GoPro to capture their activities. Most interesting ones become content on their channels.

Figure 1: Travel youtuber's moment from a video. This content wouldn't be possible without wearable cameras

Image data is the best for our use-case. So we need some kind of camera, like action cam on your head - some youtubers do this. But it is not very comfortable for ordinary people [5].

1.2 Egocentric data

I need to start by defining what type of people daily activities data is the best for my work. As I noted in above sections, in order to analyze the daily life of people, we need to record their behaviours: the places they go, the activities they perform, the people they meet, etc.
Some kind of video / photo data is better in this case, but what kind of video/photo? From some cctv cameras on the wall? It is not practical - we need to gather data per person and this approach will inevitably fail. So we need to get **egocentric data** - all a person do in image/video form in first-person perspective. So lifelogging using wearable cameras it is!

Figure 2: Action camera head mount

1.2.1 Wearable cameras

Gathering egocentric data using wearable cameras is one of the most comfortable and unintrusive ways to do lifelogging: small camera doesn't require attention during a day - and do all the job, leaving a person focused on daily events. Thus giving more reliable and accurate data. This days a small wearable camera isn't something uncommon as I noted above, people do live-streams, vlogging and photo capturing with wearable digital cameras. Our goal is a little different though.

Not all of the cameras are comfortable for the user, making difficult the recording of daily activities. Moreover, most of the cameras record videos with high frame resolution. Thus, the battery do not last for the whole day, not allowing to record for several hours daily activities. For example if a person will wear GoPro on their head like youtubers do (see Fig. 2) it will attract too much unwanted attention from others during daily activities and

Figure 3: Examples of Narrative Clip 2 camera in the wild

will inevitably affect results. So camera must be smaller, or better - less noticeable and it must be mounted in a more humble way.

While video sounds better for our goal - 6-10 hours of video will require more expensive and bigger equipment (better camera's battery, more memory) and transfer of that large amount of data will be a problem because processing this data on user's side is not always possible. Therefore, egocentric photostreams suit better our goal of analyzing long periods of time for getting understanding of the user's daily life. By photostream I mean a series of photos made one after another after some reliable period of time. Obviously less this period is the better for us. But if it is too small - same limits as described above for video - will apply.

In this study, we capture egocentric photostreams with the **Narrative Clip 2** camera (Figure 3). The pictures were recorded by several persons who used this device for 2-3 weeks each.

This camera answers the basic requirements needed for this project: It is small, not intrusive, and can be mounted on clothes.

1.3 Motivation

The contributions of this work are two fold.

- On one side, we perform an analysis of a new and not yet explored egocentric dataset. This dataset describes peoples daily activities.
- We propose the first model that addresses the analysis of routines from this kind of data. To do so, we use a pre-trained CNN as features extractor to describe the recorded data.
- Also, we evaluate the performance of the proposed method over semantic and global features

1.4 Memory structure

The following sections are organized as follows:

- **Section 2** describes related works, approaches and details literature read.
- **Section 3** details the proposal model. In this section, I explain what initial data I need, how I going to process it and how I will obtain my final results.
- **Section 4** contains all the performed experiments, from initial data analysis up to final results in table form.
- **Section 5** is my results analysis, best and worst approaches and their possible explanations, my opinion.
- **Section 6** describes the conclusions that I made of the different points treated along this work.
- **Section 7** proposes future research plans for the topic of routine recognition.
- **Section 8** is annex, where all my work related data is. Complete tables, work environment setup, etc.

2 Literature

Before diving deeper into my proposal I will mention some related papers and works I read. As I mentioned earlier, nowadays, Convolutional Neural Networks is a hot topic within computer vision so there are many works and areas where it has been applied and that could be explored. However, human activity recognition related works are particularly interesting for me. Egocentric vision is a growing area within the computer science field and that is why it got my attention. In this section I will give insight of the field of CNNs, and how they have been applied on the egocentric vision field.

2.1 Deep learning for images classification

Convolutional Neural Networks (CNNs) is the key concept that I wanted to touch in this work. My aim was to learn about how it works and to apply it to a real life problem. Why CNNs? Because CNNs have outperformed traditional classifiers on most of the tasks they have been dealing with, specially on images classification and description[6].

Lets start by defining some concepts that I learned and I will be using through this document - What is a neural network? NNs were inspired by how human brain's neurons work. Network consists of layers of neurons, where every neuron in a layer n is connected to all the neurons of layer n-1. It receives signal from each neuron from previous layer and produces a single output signal that is sent to next layer. The idea is that strengths of connections between neurons are learn-able and control strength of influence of one neuron onto another. This means every neuron has "weights" - multipliers of incoming signals, each one for a $\mathbf{l_{n-1}}$ neuron, this weights can be learned. After processing all incoming signals it sends one output. So basically: neuron performs dot product of inputs and corresponding weights and then applies non linear activation function to determine output.

Neural network is formed of layers, input layer, output layer - and certain number of hidden layers in between. Whole network represents single differentiable score function: from raw data on one end to class scores on another.

Convolutional Neural Networks are designed to work with images. It still has some layers from ordinary NN (now called fully connected layers) But the core is Convolutional layer. Key concept of it is: every neuron is connected to a limited number of closely situated neurons of previous layer. So weights of a neuron become convolutional filters and output is response to them in this spatial position.

Here are some notable architectures that can be found in the literature:

VGG network architecture, introduced by Simonyan and Zisserman [7] in 2014. This is a simple network, it has 3x3 convolutional layers stacked in increasing depth ans uses max pooling to reduce volume size. In the end it has 2 fully connected layers with 4096 nodes each, followed by softmax classifier. There are 2 variations: VGG16 and VGG19, number in the end is the amount of weight layers. Training process was challenging. Zisserman and Simonyan trained smaller network first (first layers until they converged) and used them as initialization for deeper network.

ResNet was introduced by Kaiming He et al.[8] in 2015. It's architecture is quite different: it relies on micro-architecture modules - "network-in-network architectures". Training now is much easier - even with increased depth of network (up to 152 layers in original paper). It this work we rely on a fine-tuned resnet50, a version with 50 layers that was introduced in [9].

Inception V3 was first introduced by Szegedy et al[10] in 2014. It computes 1x1, 3x3, 5x5 convolutions and acts as multi-level feature extractor. Originally it was called GoogLeNet, but next versions were renamed to Inception V[version]. it has smaller weights in comparison to ResNet and VGG. **Xception** is extention to Inception architecture, proposed by Keras library author François Chollet[11]. Key difference - it has Depthwise Separable Convolutions.

As I stated earlier CNNs are one of the areas of recent breakthroughs in computer science. New works and methods are created monthly. And it is needed to mention some state of the art performance achieved.

Image classification accuracy or, better, its inverse value - error rate is usually measured on big public Image sets, like MNIST database[12] or CIFAR-10 dataset [13] and others.

Here are some impressive results:
DenseNet achieves 3.46% error rate on CIFAR-10 [14]
ShakeDrop regularization method that can be used on ResNet like architectures achieved 3.41% on CIFAR-10 and 14.9% on CIFAR-100 [15]
CNN used in [16] achieves 3.54% error on CIFAR-10. And boosts performance up to 2.89% using the Cutout regularization technique.

2.2 Works on egocentric data

The work by Lara Oscar D et al.[17] is about general design of Human Activity Recognition (HAR) systems using egocentric data from wearable sensors to collect data and activity set to recognize up to most important classifiers in HAR. Chapters about evaluation of such systems helped in my final stages of work, particularly, for accuracy representation (Section 4.6). There wasn't much about Neural Networks - only some comparison of performance but this paper gave general impression of Activity recognition systems.

On egocentric topic. Yang Shen et al.[3] talks about activity prediction using egocentric video. Not exactly my problem, they focused mainly on LSTM (Recurrecnt Neural Network) but how they worked with time (video) is impressive.

On the topic about people's routines in context of computer science: Nicola Banovic [18] talks about extraction of routine behaviours from large behavior logs, with no ground truth defined. I have some sort of similar situation. Logs(images) I work with is quite huge and ground truth on picture level is not defined.

Activity recognition with CNNs: Egocentric activity recognition without powerful graphics card farms and all their tensor cores - but with all the limitations of wearable devices is explained by Possas et al. in [19] They achieve state-of-art performance while save energy by trading off vision based activity recognition with low power motion based sensor.Not only CNNs methods were used in this work. CNN achieved 70% accuracy

In [20] Song et al. use CNNs and other methods like Long Short Term Memory (LSTM) to boost performance for egocentric activity recognition. Multi-stream CNN + Trajectory + Fisher Vector achieved 78.4% accuracy on their Multimodal Egocentric Activity dataset.

All my work is based on CNN that was used by Cartas et al. in [9]. Authors propose ensemble classifier com posed of CNN and random forest (RF) that acts at image level. Random forest takes as input one or more concatenated output vectors from the final layers of a CNN. To improve classification performance they extended the ensemble architecture to take into account the temporal information from neighbor frames. Single prediction for a batch. Then they introduced LSTM on top of ensemble previously described to further improve this approach. In their work, they achieve with a configuration of InceptionV3 + RF + LSTM a 89.85% accuracy on NTCIR-12 egocentric dataset using batch level approach.

3 Modeling proposal

In this section, I describe with details my proposal for routine recognition from images.

- Data source
- Data representation - features extraction
- Similarity computation among days
- Classification, looking for outliers (novelty detection)

3.1 Data source

Initial data will be obtained collecting photostreams by 7 users who wore the Narrative clip 2 cameras for long periods of time. All user's images are grouped into days by capture time attribute. For convenience - all photos made from 0h 0m up to 23h 59m of one day are grouped together. This is not the best way to divide data, for example user can have activities after 12h pm, but it is the easiest.

3.2 Data representation, feature extraction

Now we need a way to extract valuable information from large amount of photos. Doing this manually obviously is not an option. According to many studies, such as the one presented by A. Krizhevsky et al.[6], applying Convolutional Neural Networks is a good way of solving this problem.

Routine is commonly describe by how sequences of activities are present in the day of people. Therefore, we propose to consider the translation of sequences of images to sequences of activity labels for their later analysis. To do so, we make use of the CNN introduced in [9]. I used resNet50 version.

It is 50 layer Resudial Network with softmax classifier at the end. This network was pre-trained on ImageNet using the Keras framework [21]. Then it was fine tuned in two phases. The last fully-connected layer was optimized in the first phase using SGD for 10 epochs for all folds, a learning rate $\alpha = 1 \times 10^{-3}$, a batch size of 32, a momentum $\mu = 0.9$, and a weight decay equal to 5×10^{-6}. During the last phase, the last residual block was also optimized using SGD with same learning rate and a batch size of 10 for three additional epochs.

Activities to recognise Since I am using first-person image data - not all activities can be reliably detected. For example camera on person's neck pointing at front will never catch it's user's mobile phone conversation. So we need a set of activities, detectable on photos with first person view. CNN must be pre-trained to be able to detect these activities.

3.3 Routine Discovery

The proposed model of routine discovery is based on the analysis and distance computation of semantic and global features describing the users recorded days. Below, I describe the process that we propose for the analysis of the problem.

3.3.1 Similarity computation among days

Using routine definition given in [1] I must find a way to tell if a user's day is similar to some others - or is it different. 'Day' means set of images made in time period of 24 hours, or, after CNN prediction application - set of labels with timestamps made in 24 hours.

Side note: distance vs. similarity. This two concepts are quite close to each other. But I need to clarify difference: Distance is "how close two objects are" - so, in numbers, 0 means objects are identical, and the higher value is - more different objects are.

Higher similarity value, on the other hand, means that objects are nearly identical, and lower value is - more different the objects are.

It is important for next steps since some clustering algorithms require one and not another.

I need to get a numerical value that represents how close a couple of days are. So some function that calculates distance between vectors, or matrices is needed, but work is complicated by the fact that days consist of different numbers of images. So comparison of data with different sizes is not trivial.

3.3.2 Data aggregation

Before jumping in and calculating all distances between all days I have a problem: Days are not of the same size so direct comparison is not possible. I need some better ways to group all labels of the days - into more comparable forms. So my ideas are:

Easiest one - throw away time values. It means get all labels into one array, every position of this array is category and its value is total occurrences of that category. Basically we have an **histogram**. Since all days numbers of images are different - it is better to normalize histograms before proceeding with comparison. Benefits of this approach is simplicity - it is very fast and easy to work with normalized histogram.

On the other hand losing huge chunk of data - time - can cost us accuracy - for example if all user's days are quite similar in terms of activities and one particular day has unusual order of them - we clearly have non routine (in context of this user) - but we won't be able to detect it with histogram approach.

Another way to group images from a day would be a huge list (or 2d array if we are using prob. vectors) of values where every entry is activity data from 1 image. Problem with this approach is that we have "holes" in day logged time - periods where camera was off. Even if two days are quite similar, if they have some non matching periods of camera being off - we can't really compare them with this approach. So we can **extend** this arrays by putting "zeros" in missing periods - about 2 zeros per minute - but it will still create us a problem - how do we compare some data to zero?

Another way to group data is to create an array of **fixed size**, for example divide a day into n-hours long periods. Activity data from all images made inside particular n-hours period will be grouped into one array entry - we can use different ways to group this data - median value, average(for prob vectors only), max(for prob vectors only), etc. if no data is present for particular period - we will write zeros there.

With this approach time data is conserved, but we still have a problem comparing "zeros" to "something". And if user have even 1 hour shift in all daily activities routine - we have a problem.

Final way to group images I came up with - is to load all data into 1 array like before, but keep indexes of this array - where logging was interrupted. Not all day comparison methods would work with this approach and one that work will require some modifications. Now after I have ways to aggregate day data - I can actually start working with similarity/distance.

3.3.3 Distances

Due to the goodness of the results achieved by the following standard two distances and similarity: *Euclidean*, *Manhattan*, and *Cosine Similarity* , we use them to compute the distance among the obtained feature vectors. When comparing matrices, we calculate a norm of difference of this matrices.

If we treat $m \times n$ matrix as $m \bullet n$ vector we can use familiar vector norms and its derivatives: Frobenius matrix norm also known as euclidean norm:

$(\sum_{i,j}(a_i^j)^2)^{0.5} \quad a_i^j \in M_1 - M_2$

$\| \cdot \|_1$ norm is basically a vector 1-norm: $\sum_{i,j} \mid a_i^j \mid \quad a_i^j \in M_1 - M_2$, Usually called Mahnattan distance.

Cosine similarity is just a cosine value of an angle between two vectors in n-dimensional space, so if vectors have same direction (but not same length) - this value will be 1, and if vectors are perpendicular - value will be 0. Cosine similarity can easily be converted to "**cosine distance**" by calculating arccos of its value - thus obtaining value of an angle between two vectors. It's only downside is that we can't calculate cosine similarity between some vector and zero-vector.

3.3.4 Grouping routine vs non routine

How exactly do we detect a routine? Without any information of a user except the one obtained in steps above? We have distances (or similarity values) between days - how do we determine if a day is ordinary or not? The idea is: since we know distances for each day to others - routine days would form some kind of clusters and non routines would be outliers!

Since this work is the first attempt to discover patterns in this dataset, we evaluate several ways of grouping the data. We implement unsupervised clustering because our aim is to find groups of similar days representing routine, and groups of not related days representing the non-routine days. In this owrk we evaluate 3 different ways of clustering:

DBSCAN is Density-Based Spatial Clustering of Applications with Noise [22]. I used Sklearn's implementation [23] - it has two important parameters epsilon (ϵ) and minimal number of samples(lets call it n for comfort). First it calculates sample's ϵ-neighborhood:

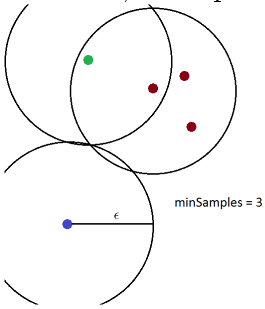
$$N_\epsilon(p) := \{q \in D \mid d(p, q) \leq \epsilon\}$$

If neighborhood has $\geq$ n points - lets call this type of sample a core-sample -all samples in distance epsilon from core samples are border samples and are grouped into same cluster. All core samples if trey are in neighborhood of another ones - are grouped into same cluster. If a sample doesn't have core samples nearby it is marked as an outlier. See figure 4. DBSCAN works great for huge datasets and explicitly shows outliers

Spectral clustering is called that way because it uses eigenvalues (spectrum) of similarity matrix to perform dimensionality reduction. It takes similarity matrix, or computes it from raw data, then calculates normalized Laplacian matrix from it:

$$L := D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$$

Figure 4: My graph of DBSCAN algorithm: red points are core samples. green one is border point of cluster, Blue point is outlier



Where A is adjacency matrix (adjacency values for different vertices in matrix form) - and D is degree matrix (edge terminations in a vertex) . Both are obtained from initial similarity matrix. Then a standard clustering method is performed over eigenvectors of Laplacian matrix. I used sklearn's implementation based on [23, 24, 25, 26]

Local novelty detection this is a bit different, I implemented it myself based on paper **Local Novelty Detection in Multi-class Recognition Problems**[27]. It is only partial implementation though. Main idea is that novelty can be inferred locally for each sample by only considering the most similar samples. So we use k closest neighbours, determined by selected distance function, to determine if this sample can belong to cluster or is it outlier.

Figure 5: My graph of proposed local novelty algorithm: we only use k nearest members connected by dotted lines of each test sample

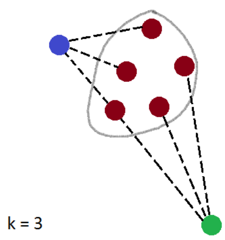
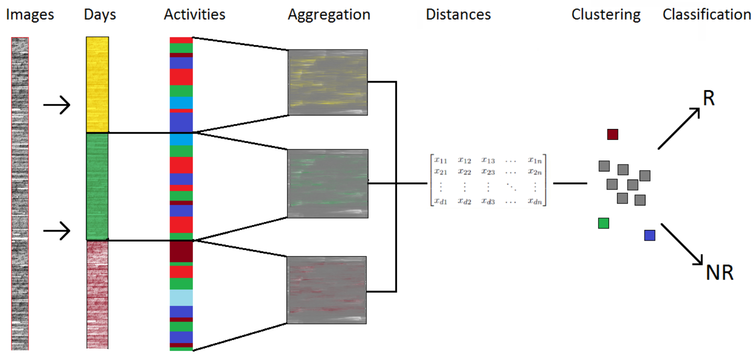


Figure 6: Proposed methodology steps: Step 1 day grouping, step 2 feature extraction, step 3: aggregation, step 4: distance calculation, step 5: clustering, step 6: final classification



4 Experiments

In this section I will explain all the work I did. From dataset analysis and days comparison to results evaluation.
Before I start I have to mention that almost all my work was done using python as programming language and jupyter notebook as editor. Ease of use of python helped me concentrating on my goals instead of programming problems and jupyter notebook helped keeping all my experiments in one place. Complete list of python packages used in this work can be found in annex. All tables in this section are reduced to its most important parts if needed. All complete tables are in annex.

4.1 Dataset

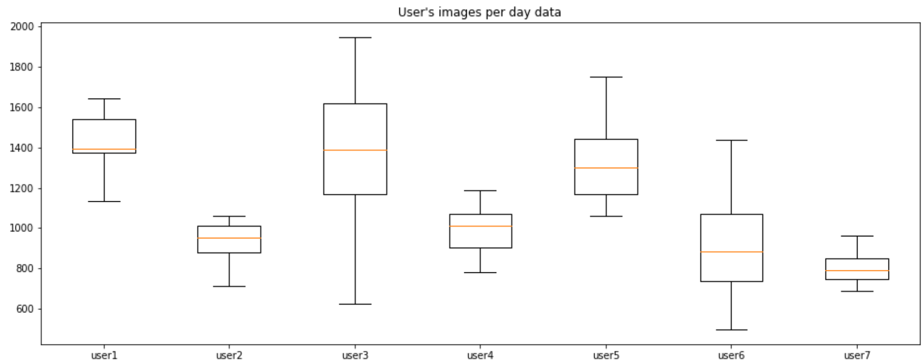
For convenience, and to save months of collecting that amount of data myself - I was provided with **University of Barcelona egoRoUB dataset**. About 70GB of egocentric photos. My first step was to study given data.

4.1.1 General statistics

egoRoUB contains 106 logged days from 7 different users. Users are simply named user1, user2... user7, days are named by its user plus consecutive number and are not in chronological order.
First, I counted days and images for each user (Fig. 7):

Since we can't really compare days from different users - some of them are in disadvantage, because fewer number of samples generally means less

Figure 7: Box plot for all users and number of images per day



user	Days	Total images	Avg. imades per day
user1	15	20699	1379.933333
user2	10	9583	958.300000
user3	16	21606	1350.375000
user4	21	20283	965.857143
user5	13	17046	1311.230769
user6	18	16592	921.777778
user7	13	10957	842.846154

Table 1: First look at egoRoUB dataset

reliable accuracy value is.

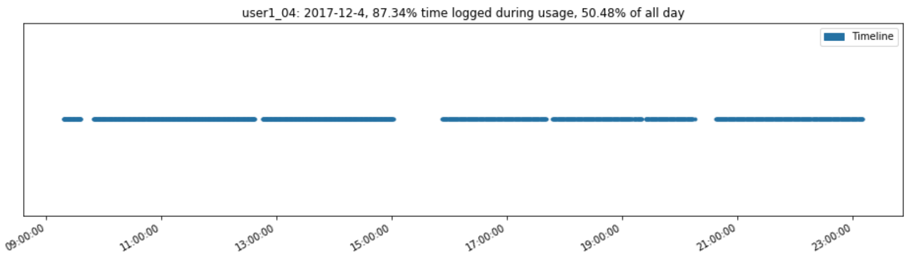
4.1.2 Daily timelines

Every photo has exact timestamp when it was taken - in its EXIF, or even in file name. So another idea I came up with was to plot every photo as a dot in simple graphic. See figure 8.

Result shows that this day's photostream is not continuous. There are multiple various length moments with no data. It is understandable - user wouldn't want camera working in certain situations.

I needed to turn this into numbers. Since camera does about 2 frames per minute (exact framerate varies though) - it is easy to calculate how many time of the day is covered. I made it a little more complicated: I calculated exact average time between all consecutive photos - if every couple of them was made in interval less than a minute. Another metric I wanted to add is logging session 'completeness' - it is percentage of time covered by 'two

Figure 8: User1 day 4 timeline representation



photos per minute' from first photo of a day to last - if interval between consecutive photos are greater than certain threshold (about one minute) - I consider that a gap. So I calculated these 2 new stats for all users:

	Avg. day time logged(h)	Avg. day-logging completeness(%)
user1	11.276275	68.221767
user2	8.443931	66.813131
user3	7.971383	70.053005
user4	8.340535	70.289946
user5	11.554292	74.588557
user6	6.595875	68.873046
user7	7.504712	81.880161

Table 2: Average hours logged and log completeness per user

I'd argue that completeness is even more important than hours logged - because hours-large holes of unknown data really complicates days comparison.

4.2 CNN

For my work I used a pre-trained resNet50 CNN introduced in [9]. It was trained to classify image as one of 21 categories:

Public Transport	Eating together	Working
Driving	Socializing	Cooking
Walking outdoor	Attending a seminar	Shopping
Walking indoor	Meeting	Talking
Biking	Reading	Resting
Drinking together	TV	Mobile
Drinking/eating alone	Cleaning and chores	Plane

All this categories are adequately identifiable on first person view photos. Also I need to add that CNN was trained to label images with single category. Multy categories are not considered. Net returns probability vector with chances of image to belong to one of the activities above.

First of all I needed to evaluate net's accuracy. To do this I was provided with ground truth.
Ground truth file contained various users - most of them was not in mine dataset. Only user1, user2 and user5 were in this file. Another, and a lot more important detail was that ground truth was given in another format, with 44 categories, not 21 used in my project.
First thing I had to do is "translate" their 44 categories into 21 hours. It was not always possible - for example we don't have equivalent for "Meditating" or "Praying" or "Writing". Other categories like "Bus" or "Metro" was translated into "Public transport"
How I calculated accuracy:
- only user1, user2 and user5 data was used
- only "translatable" categories were used, other data was discarded

Results:
Accuracy for user1: 40.51%
Accuracy for user2: 50.44%
Accuracy for user5: 41.13%
Total accuracy: 42.62%
Total elements in ground truth: 40991

That is quite less than I expected, even with foreign ground truth. So I decided to look further into this data. For every category I calculated two most frequent incorrect predictions, here are some excerpts from results table (complete tables are in annex. See tables 13 - 15):
This looks a little better - for example biking seems close to walking outdoor (at least from 1st person point of view), or public transport can be confused

Category	Accuracy	Closest 1	Closest 2
Public Transport	27.4%	17.9% Plane	10.5% Walking indoor
Walking outdoor	75%	6.6% Mobile	4.9% Walking indoor
Walking indoor	31.4%	12.8% Shopping	11.2% Mobile
Attending a seminar	24.7%	12.8% Drinking/eating alone	9.6% Mobile
TV	47%	18.5% Drinking/eating alone	12.3% Shopping
Working	47.8%	24.1% Mobile	5.8% Shopping
Shopping	77.4%	8.7% Walking indoor	3.9% Mobile
Mobile	40.1%	10.2% Public Transport	8.1% Plane
Biking	58.4%	27.7% Walking outdoor	2.9% Driving
Plane	75.5%	9.6% Attending a seminar	6.8% Mobile

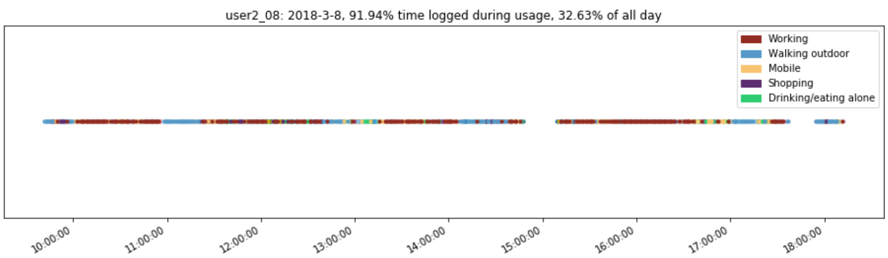
Table 3: per category CNN accuracy.

with sitting in a plane. For my work - not the dry values of accuracy are important - but assigning similar looking activities one label. This allows us to compare days successfully even if categories assigned are not exactly correct.

4.3 Labelled dataset

Now I have assigned labels to all photos in the dataset. It took some time for my humble notebook to finish calculations. Took about half an hour to chew through a day worth of photos. Now what?
First of all I needed to see how this data looks. I modified my timeline plot. I assigned a color for all of the 21 categories. I tried to assign shades of one color to similar activities for example all transportation (walking, car, public transport, plane) are shades of blue, eating activities are shades of green, etc.. 5 most frequent activities are highlighted on graphic.
Lets see how it looks:

Figure 9: User2 day 8 timeline with activities



After studying this graphics for different users and days I can tell that sequences of same activity in a row are generally correct. Also I generated tables of quantitative data of all dataset per categories, see annex.

4.4 Ground truth

Before digging deeper into days analysis it is better to define ground truth. To do this I have to manually explore day after day of photos and determine if they look like routine or no. I did this in 2 phases for user:
- Firstly a quick look on all of users days scrolling through wall of images
- Secondly I did a more detailed and slow days examination
Step 1 is needed to make myself familiar with user's activities - because what is a routine for one may be unusual day for another. This is my version of ground truth:

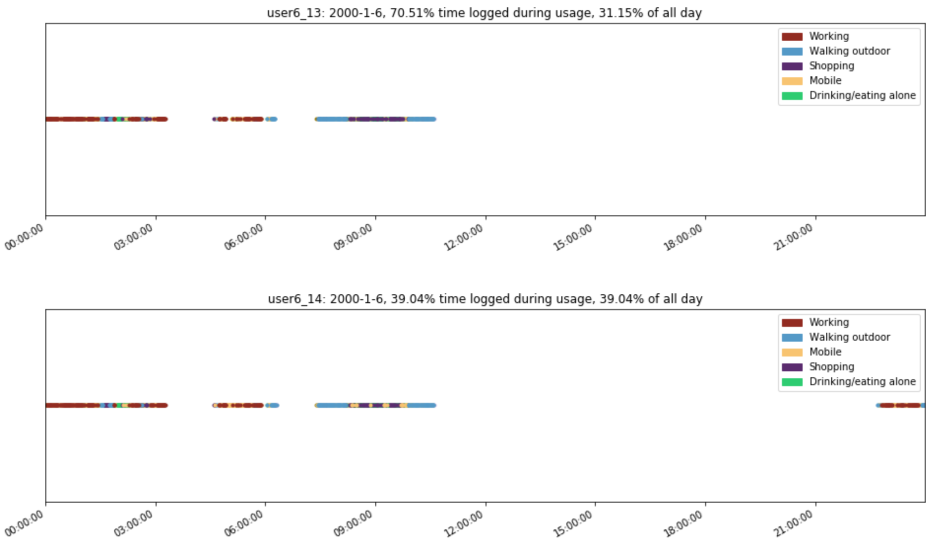
	user1	user2	user3	user4	user5	user6	user7
day 1	R	R	R	R	R	R	NR
day 2	R	NR	R	R	R	R	R
day 3	R	NR	NR	NR	R	NR	NR
day 4	R	NR	NR	NR	NR	R	NR
day 5	NR	R	R	R	NR	R	R
day 6	R	R	R	R	NR	R	NR
day 7	NR	R	R	R	NR	NR	R
day 8	R	R	NR	NR	NR	R	R
day 9	R	R	R	NR	NR	NR	R
day 10	R	NR	R	NR	R	NR	R
day 11	R		NR	R	R	R	NR
day 12	R		R	NR	R	EX	NR
day 13	NR		R	R	NR	R	R
day 14	R		R	NR		EX	
day 15	EX		R	NR		R	
day 16			R	NR		R	
day 17				NR		R	
day 18				NR		R	
day 19				NR			
day 20				EX			
day 21				R			

Table 4: Ground truth for days. "R" means routine, "NR" means not routine and "EX" means this day is excluded from study

4.4.1 Excluded days

Some days must be excluded from posterior calculations, because there is not enough data to make successful conclusions. I excluded following days:
user1 day 15: there are just 178 photos in this day, about 1.5hours of logged time
user4 day 20: there are just 273 photos in this day, about 2.2hours of logged time
user6 day 12: there are just 92 photos in this day, about 50 minutes of logged time
user6 day 14: this is unusual, there is enough data, but camera's clock was reset to January year 2000. And this day got data from 2 consecutive days. Out of 1049 photos - 837 are exact copies of day 13 - previous one, and the rest belongs to day 15, so my option was to exclue day 13 or day14, I decided that 14 is worse because it contains data from two days

Figure 10: User6 days 13 and 14 similarities



4.5 Problems with dataset

In this section I will mention some difficulties I had with given data. It is understandable given its size: 70 gigabytes, 106 days, 116766 photos.

4.5.1 Exif absence

My initial way to extract time data from photo was reading its EXIF. Main advantage of doing this is universality: Exchangeable Image File Format allows extracting data from any photo made on pretty much any modern camera.
But I quickly found out that not all photos have EXIF. For example User4 day03 - no exif data for all photos of that day. Looks like some days were logged using old version of camera, or camera's settings were incorrect.
The only way to extract time data from these photos was to read it from their file names. It works, but the cost is universality: only photos with filenames like **yyyymmdd_hhmmss_xxx.jpg** are supported - because it is really hard to write universal date-from-string parser, if you count all the ways time is written into photo names of different manufacturers...

4.5.2 Rotation problem

When I was defining ground truth I noticed that some days of user4 had images rotated 90 degrees clockwise. Some photos of a affected day were fine, but majority was rotated. Looks like some settings of the camera were set wrong, or more probably - it's gyroscope detected some orientation change. That explains why some photos, usually at the beginning, are fine
Main problem is that CNN prediction is not rotation invariant. So labels from this day may be corrupted (see Fig. 11).

Since exif was not available exactly for same days and there were no reliable manner to detect if image was flipped - the only solution I found was to rotate data from photos of selected days before executing CNN prediction. Other solution would be applying another CNN pre-trained to detect rotations, but that is not purpose of my work.

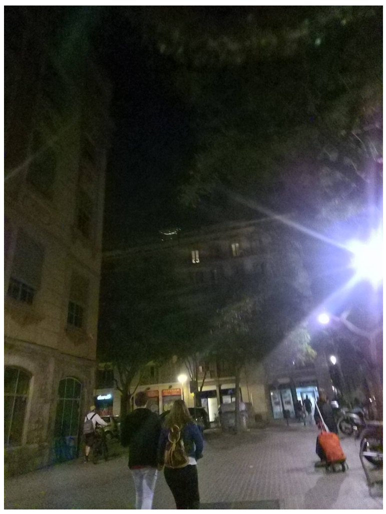
4.5.3 Wrong camera clock settings

User 6 days from 11 to 18 were logged with wrong time settings - January year 2000... while we still have data continuity - I mean we can see that photo2 was taken 30 seconds after photo1 - we lose time of the day this photo was taken - it is a problem for routine detection methods that rely closely on time - like my fixed n-row tables (see below in methods). See Figure 10 - timeline of day13. while is is shown that all the activities were from 0h00m to 10h36m - photos taken around 10am - are clearly made during the night.

Figure 11: Applying CNN's prediction to different rotations of same image

See figure 12

Figure 12: User6 day 13. Time written into filename is 10h 32m 57s am



4.6 Evaluation and metrics

After applying different methods of aggregation, distance calculation and clustering described in chapter 3, I want to receive clear classification if a day is detected as routine or no. Easiest metrics is simple accuracy - from 0% to 100% if all days are detected correctly. But not everything is that simple. From all 102 days of my dataset (4 days were excluded) 40 are classified as not routines. So if some clustering method is wrong and just marks everything as routine - we still get about 60.8% accuracy! This is unacceptable - so I have 3 accuracy values: total accuracy, accuracy for routines only and accuracy for not routines only. It also helps to detect if some methods are particularly good for detection of one but not another. To compare results with different percentages I used F-measure value for non-routines, because it is used in different works such as [28] and [17] to evaluate classification. F-measure is:

$$F_1 = 2 * \left(\frac{precision * recall}{precision + recall} \right)$$

Where precision is number of correctly detected non routines divided by all number of all samples marked as non routine and recall is non routine accuracy. I based F_1 measure on non routines because I think focus on non routine detection is more important.

4.7 Experimental setup

So now I have label, probability vector and timestamp for every photo in dataset. These photos are grouped into "days". Now what?
Now I need to consistently apply all possible combinations of methods described in chapters 3.3 - 3.4.

4.8 Feature vectors

I use labels and it's corresponding probability vectors as representation of a single image. But to take another look, and obtain another source of data I decided to take output of penultimate layer of CNN - it returns a 2048 feature vector that contains features of processed image. Obviously all calculations with 2048 vectors would take much, much longer, but it gives us another comparison item!

4.8.1 Image's data aggregation

I used 4 ways to group inconsistent day images data into one array:

- **Histograms** - for their ease of use and work with.
- **Extended tables** - my idea was to fill all the gaps in lifelogging process with zeros - all missed time between photos is filled with zero data - 2 zeros per minute (or rows of zeros if we work with matrices). Resulting data still has inconsistent length but at least we can compare it.
- **Blocks table** - modification of previous method, but now I treat all uninterrupted data as one array, so a day is represented as a group of such arrays - group's size is variable: from one element if the day has no interruptions - to many. All comparison of these days must be inside this smaller arrays.
- **Fixed size tables with partial time data conservation** - I divided 24 hours of a day into 48 equal intervals (30 minutes each) Then I grouped all image data inside one interval - into one entry. There are many ways to do this. Like take median value, or average or even max (for probability vectors only). I used max value per position with probability vectors. In the end I got 48-row matrix with mostly conserved absolute time values for activities (within a day). I'll refer to this method as **fixed size tables** in my work.

4.8.2 Distance between days, similarity

For every aggregation technique I calculated distances between days.

Histograms. I calculated all 3 distances I mentioned in chapter 3.4. Nothing unexpected.

Extended tables here it becomes more complicated - tables have different size but distance calculation functions require it to have the same. Solution I came up with was - apply smaller matrix as sliding window over bigger one - using all 3 distances - and take the lowest result obtained. But what to to with the rest of bigger matrix data? We have 2 approaches:

- Optimistic - we assume that all missing data from smaller matrix is exactly the same as the rest of a bigger one - and we just return smallest sliding window result.

- Pessimistic - we assume that missing data is different and to penalize we add norm of unused bigger matrix parts to sliding window result. Note that this approach doesn't work with cosine distance because it cant take zero vectors.

Blocks tables - took the same approach as Extended tables. I used sliding window, but only within its blocks: It is like tetris - I take first block of day1 and try to "fit" it inside all blocks of day2. Take the best result sliding window gives and mark sections of both days as already processed. Repeat until one of the days is completely "fit" into another. Return accumulated result. Same pessimistic and optimistic approach as in previous method applies.

Fixed size tables - since data size is fixed - there should be no problem applying all three distance calculation functions.

4.8.3 Clustering, Different configurations

I calculated distance matrices for every user using all combinations previously explained methods. Its time to apply clustering.

DBSCAN has two parameters. epsilon and minimal number of samples. We can't really compare users, so clustering must be performed per user - and we have quite small number of days per user - on average 14, ranging from 10 to 20. So i decided to keep min.samples small - I chose 3. I tested other numbers and didn't get much better results due to small sample numbers. Epsilon on the other hand was really hard to choose. With right epsilon value accuracy can be really high, but main problem is that we can't know that value a priory. And adjusting this value for every user is not a good idea. So I chose epsilon in function of average day-to day distance per user. I used different fractions of average distance in my final accuracy calculations.

Spectral clustering unlike other two clustering methods rely on similarity matrix, not distance matrix. So I need to convert my distance matrices. Do to this, I used Gaussian Kernel: $a_i^j = e^{{a_i^j}^2/2*d^2}$ so 0 distance gives highest value in this similarity matrix. Oh and I needed to normalize distance matrix before applying kernel because distances can be quite big so Gaussian kernel would return zero.

Local novelty In my partial implementation of [27] k is the number of closest neighbors to consider. I used three k values: k=2, k=3 and k=4. Again due to small number of samples per user I couldn't really use more.

4.9 Results

I calculated accuracy for all possible combinations of methods described above. Full results tables with total accuracy and per user data are found in annex, section 8.4. In this section for every day data aggregation method I will mention only the highest f-measure results per clustering method as explained in section 4.6.

Method	Distance	Parametrs	F-measure	Total acc.	Routine acc.	Non routine acc.
DBSCAN	cosine	eps=0.51 avg	0.7800	78.43	66.13	97.50
Spectral	cosine	-	0.4243	62.75	80.65	35.00
LN	manhat.	k=2	0.5902	75.49	95.16	45.00

Table 5: Histograms. Best results from tables 31, 32 and 33

Method	Distance	Parametrs	F-measure	Total acc.	Routine acc.	Non routine acc.
DBSCAN	cosine	eps=0.95 avg	0.5790	68.63	77.42	55.00
Spectral	euclid.	-	0.5867	69.61	79.03	55.00
LN	cosine	k=3	0.6897	73.53	72.58	75.00

Table 6: Extended tables, optimistic approach. Best results from tables 34, 36 and 38

Method	Distance	Parametrs	F-measure	Total acc.	Routine acc.	Non routine acc.
DBSCAN	euclid.	eps=0.80 avg	0.5044	44.12	25.81	72.50
Spectral	manhat.	-	0.3243	50.98	64.52	30.00
LN	manhat.	k=4	0.4691	57.84	64.52	47.50

Table 7: Extended tables, pessimistic approach. Best results from tables 35, 37 and 39

Method	Distance	Parametrs	F-measure	Total acc.	Routine acc.	Non routine acc.
DBSCAN	cosine	eps=0.93 avg	0.6250	64.71	58.06	75.00
Spectral	euclid.	-	0.4000	61.76	80.65	32.50
LN	cosine	k=3	0.6353	69.61	70.97	67.50

Table 8: Blocks tables, optimistic approach. Best results from tables 40, 42 and 44

Method	Distance	Parametrs	F-measure	Total acc.	Routine acc.	Non routine acc.
DBSCAN	euclid.	eps=0.77 avg	0.4250	54.90	62.90	42.50
Spectral	euclid.	-	0.4412	62.75	79.03	37.50
LN	manhat.	k=2	0.6046	66.67	67.74	65.00

Table 9: Blocks tables, pessimistic approach. Best results from tables 41, 43 and 45

Method	Distance	Parametrs	F-measure	Total acc.	Routine acc.	Non routine acc.
DBSCAN	euclid.	eps=0.84 avg	0.6400	64.81	54.84	80.00
Spectral	euclid.	-	0.4516	66.67	87.10	35.00
LN	euclid.	k=2	0.6500	72.55	77.42	65.00

Table 10: Fixed size tables, Best results from tables 46, 47 and 48

4.10 Feature vectors results

All operations with 2048 feature vectors took about 15 times more time than all operations with normal labels / probabilities. I will include here only final table with highest f-measure values.

Aggregation	Cluster.+novelty	Distance	F-measure	Total acc.	Routine acc.	Non routine acc.
Blocks t.	LN	man	0.7632	82.35	88.71	72.50
Histogram	LN	cosine	0.6756	76.47	85.48	62.50
Fixed size t.	LN	euclidean	0.6341	70.59	74.19	65.00
Extended t.	DBSC	cosine	0.5970	47.06	12.90	100.00

Table 11: Feature Vector Final results from tables 49, 50,51 and 52

5 Discussions

In this section some thoughts about executed experiments are described.

5.1 Dataset

In this work we found some problems/challenges on the way. On one side, the dataset was not balanced. It was not balanced in terms of activity classes nor Routine/Non-routine distribution. However it was close to what you would really obtain from people if they'll use chosen lifelogging method. Also proposed method assumes that most of user's days - or at least most similar days are routines. But what if majority of users days are classified in ground truth as Novelty samples? It is possible for us humans to assume that more "boring" days tend to be routines but what if proposed model actually finds more similarities in these "novelty" samples? For example user 5 tends to have really low accuracy values for almost all methods but from ground truth, and detailed manual day study we see that:

a) user has less than half days classified as routines.

b) majority of non-routine days are voyage to some international conference - they can be easily grouped as a cluster.

This may be the case where machine is right and we, mere humans are wrong!

Activity recognition validation problem: As seen in our experiments - CNN accuracy validation wasn't impressive. Partially it is because we used ground truth from another project given in another format and from 116766 images only about 40000 from 3 users had GT defined.

Another problem was the analysis of non informative images: activities on some photos were not identifiable, not even for humans - like all black or heavily blurred images. Not to forget rotation as described in section 4.5.2. But not all images could be rotated - some had 45^o rotation - and could not be corrected -without cropping and it would be a huge problem doing it for many samples.

5.2 Clustering experiments

Due to the specific of our work, traditional clustering methods do not achieve the better performance in this problem when using default parameters

Spectral clustering generally performed a lot worse than two others. Part of the reason is that we didn't use affinity matrices, because other methods require distance matrices and methods to create such a matrix for a day are challenge by itself - and huge part of this work - so we just converted our existing distance matrices to affinity ones using Gaussian kernel. Other reason may be is that Spectral clustering method don't explicitly indicate outliers - we considered resulting clusters with one element as novelty - but what if only non routines are put into 1 cluster? Method technically works but we have no way to tell if it is correct.

DBSCAN may perform surprisingly good and awfully bad based on initial parameters. In fact, our best classification result was obtained with this method. However in practice it is really hard to get it's epsilon value right. So this method is good if right values can be learned.

Local novelty performed quite good - it didn't reach best DBSCAN accuracy, but achieved Second-best result without the need to adjust parameters. In fact This method shows potential that needs to be studied in future developments on this project.

5.3 On aggregation methods

Histograms performed really well - Our initial assumption that absence of time data will noticeably hurt accuracy was wrong. This method has lowest computational cost.

Extended tables: "Pessimistic" approach worked significantly worse than "Optimistic" so finding similarities between days was better than differences. In general this method has highest computational cost among competitors. It is understandable giving how many operations sliding window process has to perform. On accuracy side it gave 3rd best result.

Blocks tables didn't disappoint. Having high performance cost (only extended tables were more costly) Labels accuracy was mediocre, but if we use feature vectors instead - we get second best f-measure result in whole work! And again - pessimistic approach didn't work.

Fixed size tables performed ok. Almost reached best result of Extended tables with a lot less computational cost.

5.4 Feature vectors

Feature vectors gave worse results for 3 out of 4 clustering methods (one result was really close to labels though). However their single better result is quite impressive. Only Labels + DBSCAN with right epsilon can surpass it. However computational cost of 2048-long vectors is very taxing. In fact it took about 15 more times to perform same calculations on my machine.

6 Conclusions

Our goal was to discover routine in egocentric photostreams. To this end as it was described above we collected egocentric dataset using Narrative clip 2 wearable camera, studied this dataset and extracted labels using Convolutional Neural Network resNet50[9]. We applied our proposed methods and obtained results. Now we are ready to give conclusions. To this end, we proposed a model that analyses days based on the activity probability vector extracted by a pre-trained CNNs (or we use 2048 image's features vector), aggregates this vectors to form an entity representing a single day and classifies this day as routine or no, based on clustering that was done on distance matrices of this days.

From the results we can see that the best method is **Labels + histograms + DBSCAN + cosine distance** - it has 78.43% accuracy overall and 97.5% accuracy on non-routine detection. It has 0.78 f-measure. However this method is heavy relied on its DBSCAN epsilon value, and it is not always possible to get right epsilon a priory, this value is better to be learned.

The second best result gave us usage of **feature vectors** instead of labels: **Features + Blocks + Local Novelty** achieves 0.76 f-measure with 72.5% non routine accurate and whopping 82.35% total accuracy.

Third best result is **Labels + extended matrix + Local Novelty + cosine distance**. It has 73.53% overall accuracy, and 75% non routine accuracy. It has 0.6897 f-measure. However it has highest computational cost.

From this we can conclude that the proposed model works, though there is room for improvement.

Aggregation	Data	Cluster.+novelty	Distance	F-measure	Total acc.	Routine acc.	Non routine acc.
Histogram	Labels	DBSCAN	cosine	**0.7800**	78.43	66.13	**97.5**
Blocks t.	Features	LN	manhat.	0.7632	**82.35**	**88.71**	72.50
Extended t.	Labels	LN	cosine	0.6897	73.53	72.58	75.00
Fixed size t.	Labels	LN	euclidean	0.6353	69.61	70.97	67.50

Table 12: Best results per aggregation method.

7 Future work

Finally, we propose some lines of research that we believe can help improving the performance of the proposed method:

Noise reduction: When you see a batch of predictions labels and they look something like [work, work, work, TV, work work work] - One may assume that TV labelled activity is probably marked wrong. If we treat array of labels from photos of a day like some kind of image - we can perform some analog of median filter for photos. it will help us to get rid of unidentifiable photos without having to mess with dataset.

Block table (see section 4.8.1) comparison rework. While blocks tables performs surprisingly well for feature vectors, we suppose that they still can be improved: in current version comparison is done the way that it tries to compare earlier time blocks with earlier time blocks first. Then the rest of the blocks are "fit" where they can. So proposed change is: compare biggest blocks first - without breaking them. We believe this can improve days comparison.

We adapted the method proposed in [27] to our problem. However, future steps can go on the development of new techniques of Local Novelty detection. Several other studies can be implemented to study their performance when addressing routine recognition.

8 Annex

In this section we include and list all the tables generated for the dataset description and later analysis.

8.1 CNN accuracy tables

Tables of accuracy evaluation per category. Closest 1 and Closest 2 are accuracy for the most frequent categories that was confused with the main one

stat	Accuracy	Closest 1	Closest 2
Public Transport	23.5	18.0% Plane	15.0% Walking indoor
Driving	NaN	NaN	NaN
Walking outdoor	73.9	9.5% Mobile	5.8% Walking indoor
Walking indoor	29.6	16.2% Shopping	11.4% Mobile
Drinking together	2.9	45.9% Shopping	21.8% Mobile
Drinking/eating alone	20.6	20.8% Working	16.1% Shopping
Eating together	11.4	23.4% Shopping	13.2% Socializing
Attending a seminar	11.2	46.1% Working	13.5% Socializing
Meeting	9.2	30.1% Talking	24.2% Shopping
Reading	6.1	66.7% Shopping	15.2% Drinking/eating alone
TV	0	50.0% Walking outdoor	50.0% Walking indoor
Cleaning and chores	15.7	28.4% Cooking	23.5% Mobile
Working	46.6	31.9% Mobile	8.9% Shopping
Cooking	6.2	18.8% Walking indoor	18.8% Talking
Shopping	84.6	3.6% Mobile	2.0% Walking indoor
Mobile	39.9	14.4% Public Transport	8.2% Walking indoor
Biking	NaN	NaN	NaN
Socializing	NaN	NaN	NaN
Plane	NaN	NaN	NaN

Table 13: User1 CNN accuracy

stat	Accuracy	Closest 1	Closest 2
Public Transport	45.8	20.3% Plane	8.5% Walking indoor
Driving	NaN	NaN	NaN
Walking outdoor	77.8	5.6% Walking indoor	4.5% Mobile
Walking indoor	38	10.3% Mobile	9.8% Walking outdoor
Drinking together	13.3	66.7% Shopping	20.0% Mobile
Drinking/eating alone	25.9	17.2% Mobile	9.2% Walking outdoor
Eating together	11.2	49.0% Shopping	10.2% Drinking/eating alone
Attending a seminar	NaN	NaN	NaN
Meeting	NaN	NaN	NaN
Reading	20	60.0% Walking outdoor	20.0% Mobile
TV	0	100.0% Walking outdoor	0.0% Plane
Cleaning and chores	12.5	33.7% Driving	13.5% Mobile
Working	52.9	20.7% Mobile	4.5% Walking indoor
Cooking	5.4	16.3% Shopping	15.7% Eating together
Shopping	71.4	11.3% Walking indoor	6.7% Mobile
Mobile	45.7	18.2% Walking outdoor	7.6% Working
Biking	58.4	27.7% Walking outdoor	2.9% Driving
Socializing	NaN	NaN	NaN
Plane	NaN	NaN	NaN

Table 14: User2 CNN accuracy

stat	Accuracy	Closest 1	Closest 2
Public Transport	28.7	17.2% Plane	7.9% Reading
Driving	NaN	NaN	NaN
Walking outdoor	73.6	5.7% Mobile	3.3% Walking indoor
Walking indoor	31.2	11.7% Shopping	11.3% Mobile
Drinking together	22.3	20.6% Drinking/eating alone	14.3% Shopping
Drinking/eating alone	16.1	30.9% Plane	19.4% Shopping
Eating together	8.8	21.0% Shopping	20.5% Drinking/eating alone
Attending a seminar	26	14.0% Drinking/eating alone	10.3% Mobile
Meeting	16.2	13.7% Walking indoor	13.4% Shopping
Reading	23.5	25.4% Walking indoor	9.2% Mobile
TV	47.4	18.6% Drinking/eating alone	12.4% Shopping
Cleaning and chores	12.7	20.0% Walking indoor	10.9% Resting
Working	47	10.7% Attending a seminar	9.7% Walking indoor
Cooking	15.8	26.3% Walking indoor	15.8% Walking outdoor
Shopping	76	12.6% Walking indoor	3.9% Socializing
Mobile	36	15.9% Plane	10.4% Public Transport
Biking	NaN	NaN	NaN
Socializing	0	53.2% Mobile	21.1% Reading
Plane	75.5	9.6% Attending a seminar	6.8% Mobile

Table 15: User5 CNN accuracy

stat	Accuracy	Closest 1	Closest 2
Public Transport	27.4	17.9% Plane	10.5% Walking indoor
Driving	NaN	NaN	NaN
Walking outdoor	75	6.6% Mobile	4.9% Walking indoor
Walking indoor	31.4	12.8% Shopping	11.2% Mobile
Drinking together	15.2	27.0% Shopping	13.8% Drinking/eating alone
Drinking/eating alone	21.2	13.7% Shopping	12.7% Mobile
Eating together	10.2	23.3% Shopping	15.1% Drinking/eating alone
Attending a seminar	24.7	12.8% Drinking/eating alone	9.6% Mobile
Meeting	12	22.1% Talking	20.0% Shopping
Reading	21.6	22.6% Walking indoor	14.5% Shopping
TV	47	18.5% Drinking/eating alone	12.3% Shopping
Cleaning and chores	13.4	19.5% Driving	15.1% Mobile
Working	47.8	24.1% Mobile	5.8% Shopping
Cooking	5.8	15.3% Shopping	14.5% Eating together
Shopping	77.4	8.7% Walking indoor	3.9% Mobile
Mobile	40.1	10.2% Public Transport	8.1% Plane
Biking	58.4	27.7% Walking outdoor	2.9% Driving
Socializing	0	53.2% Mobile	21.1% Reading
Plane	75.5	9.6% Attending a seminar	6.8% Mobile

Table 16: All users CNN accuracy

8.2 User activities stats tables

For every user in dataset I generated a table with statistical data for all 21 activities.

Day	All days			Routine			Non routine		
stat	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation
Public Transport	600.0	42.857143	20.468991	453.0	41.181818	20.590458	147.0	49.000000	18.779421
Driving	234.0	16.714286	13.904925	165.0	15.000000	14.734005	69.0	23.000000	7.483315
Walking outdoor	2039.0	145.642857	47.522336	1471.0	133.727273	45.371378	568.0	189.333333	23.697163
Walking indoor	1700.0	121.428571	26.537613	1360.0	123.636364	21.839384	340.0	113.333333	38.134266
Biking	21.0	1.500000	1.546886	17.0	1.545455	1.671343	4.0	1.333333	0.942809
Drinking together	331.0	23.642857	10.741157	214.0	19.454545	7.981383	117.0	39.000000	2.160247
Drinking/eating alone	548.0	39.142857	16.621231	358.0	32.545455	8.937802	190.0	63.333333	15.860503
Eating together	228.0	16.285714	20.242913	93.0	8.454545	11.007135	135.0	45.000000	20.461346
Socializing	436.0	31.142857	32.653718	236.0	21.454545	24.952186	200.0	66.666667	32.967998
Attending a seminar	197.0	14.071429	13.338796	142.0	12.909091	14.209181	55.0	18.333333	8.178563
Meeting	382.0	27.285714	23.149911	235.0	21.363636	20.812028	147.0	49.000000	17.682383
Reading	227.0	16.214286	10.379778	139.0	12.636364	4.829318	88.0	29.333333	14.079141
TV	153.0	10.928571	9.505369	130.0	11.818182	10.460473	23.0	7.666667	2.624669
Cleaning and chores	244.0	17.428571	7.208272	193.0	17.545455	7.377652	51.0	17.000000	6.531973
Working	4640.0	331.428571	147.831813	4096.0	372.363636	135.133787	544.0	181.333333	79.725919
Cooking	107.0	7.642857	5.959472	80.0	7.272727	6.411508	27.0	9.000000	3.559026
Shopping	2529.0	180.642857	95.306728	1742.0	158.363636	44.837632	787.0	262.333333	162.862587
Talking	947.0	67.642857	35.600175	681.0	61.909091	34.344391	266.0	88.666667	32.045107
Resting	316.0	22.571429	8.077431	244.0	22.181818	8.244207	72.0	24.000000	7.257180
Mobile	4285.0	306.071429	143.140023	3676.0	334.181818	148.477619	609.0	203.000000	35.505868
Plane	357.0	25.500000	16.838519	281.0	25.545455	18.082456	76.0	25.333333	11.145502

Table 17: User1 activities statistics

Day	All days			Routine			Non routine		
stat	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation
Public Transport	229.0	22.9	16.139703	118.0	19.666667	9.445752	111.0	27.75	21.867499
Driving	242.0	24.2	10.842509	139.0	23.166667	13.158225	103.0	25.75	5.494315
Walking outdoor	2026.0	202.6	60.998689	1041.0	173.500000	42.625305	985.0	246.25	58.319701
Walking indoor	687.0	68.7	37.427396	484.0	80.666667	44.116009	203.0	50.75	6.759253
Biking	365.0	36.5	24.980993	124.0	20.666667	13.535960	241.0	60.25	18.579222
Drinking together	153.0	15.3	9.077995	119.0	19.833333	8.858455	34.0	8.50	3.354102
Drinking/eating alone	408.0	40.8	16.209874	268.0	44.666667	17.857460	140.0	35.00	11.067972
Eating together	152.0	15.2	8.009994	69.0	11.500000	5.737305	83.0	20.75	7.725769
Socializing	270.0	27.0	24.012497	199.0	33.166667	29.151425	71.0	17.75	4.918079
Attending a seminar	69.0	6.9	9.027181	48.0	8.000000	10.066446	21.0	5.25	6.869316
Meeting	207.0	20.7	7.963040	138.0	23.000000	4.546061	69.0	17.25	10.377259
Reading	102.0	10.2	7.560423	60.0	10.000000	5.416026	42.0	10.50	9.937303
TV	156.0	15.6	14.887579	125.0	20.833333	16.506733	31.0	7.75	6.533567
Cleaning and chores	141.0	14.1	6.992138	73.0	12.166667	5.013870	68.0	17.00	8.396428
Working	1581.0	158.1	104.621652	1223.0	203.833333	105.830709	358.0	89.50	52.160809
Cooking	71.0	7.1	4.346263	31.0	5.166667	3.435921	40.0	10.00	3.937004
Shopping	737.0	73.7	72.437628	534.0	89.000000	88.498588	203.0	50.75	22.185299
Talking	252.0	25.2	13.724431	179.0	29.833333	6.986097	73.0	18.25	17.809759
Resting	195.0	19.5	8.357631	98.0	16.333333	4.678556	97.0	24.25	10.207228
Mobile	1452.0	145.2	67.139854	993.0	165.500000	66.284111	459.0	114.75	55.979349
Plane	88.0	8.8	9.260670	35.0	5.833333	3.804237	53.0	13.25	12.636752

Table 18: User2 activities statistics

Day	All days			Routine			Non routine		
stat	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation
Public Transport	568.0	35.5000	28.220117	345.0	28.750000	14.623754	223.0	55.75	44.689904
Driving	99.0	6.1875	6.866756	85.0	7.083333	7.521728	14.0	3.50	3.041381
Walking outdoor	1510.0	94.3750	71.565420	1004.0	83.666667	41.041307	506.0	126.50	118.563274
Walking indoor	1527.0	95.4375	34.447367	1231.0	102.583333	26.310512	296.0	74.00	45.354162
Biking	8.0	0.5000	0.790569	7.0	0.583333	0.862007	1.0	0.25	0.433013
Drinking together	384.0	24.0000	21.900342	302.0	25.166667	20.090766	82.0	20.50	26.291634
Drinking/eating alone	1166.0	72.8750	39.815002	923.0	76.916667	42.956292	243.0	60.75	24.681724
Eating together	488.0	30.5000	22.937415	413.0	34.416667	20.734264	75.0	18.75	25.113492
Socializing	330.0	20.6250	19.006167	214.0	17.833333	9.745369	116.0	29.00	32.657312
Attending a seminar	483.0	30.1875	39.633033	420.0	35.000000	44.183330	63.0	15.75	12.193748
Meeting	943.0	58.9375	32.621827	843.0	70.250000	29.889031	100.0	25.00	6.403124
Reading	2053.0	128.3125	92.684761	1960.0	163.333333	80.732411	93.0	23.25	9.549215
TV	91.0	5.6875	6.573800	64.0	5.333333	4.478343	27.0	6.75	10.544548
Cleaning and chores	462.0	28.8750	20.152156	350.0	29.166667	20.895906	112.0	28.00	17.705931
Working	5028.0	314.2500	157.190768	4408.0	367.333333	136.465095	620.0	155.00	95.673925
Cooking	183.0	11.4375	7.373676	144.0	12.000000	8.195527	39.0	9.75	3.491060
Shopping	2159.0	134.9375	65.282529	1915.0	159.583333	55.481916	244.0	61.00	22.880122
Talking	659.0	41.1875	19.452567	579.0	48.250000	15.454908	80.0	20.00	14.089003
Resting	232.0	14.5000	9.034655	167.0	13.916667	5.090159	65.0	16.25	15.642490
Mobile	3028.0	189.2500	77.680033	2487.0	207.250000	64.383131	541.0	135.25	88.392237
Plane	205.0	12.8125	8.164089	142.0	11.833333	6.986097	63.0	15.75	10.425330

Table 19: User3 activities statistics

Day	All days			Routine			Non routine		
stat	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation
Public Transport	413.0	20.65	23.234188	116.0	14.500	14.186261	297.0	24.750000	26.898343
Driving	788.0	39.40	43.677683	197.0	24.625	47.652223	591.0	49.250000	37.725157
Walking outdoor	3894.0	194.70	131.022555	1075.0	134.375	60.280464	2819.0	234.916667	148.813899
Walking indoor	3287.0	164.35	93.846297	1178.0	147.250	112.514166	2109.0	175.750000	76.902671
Biking	61.0	3.05	3.073679	32.0	4.000	3.391165	29.0	2.416667	2.660148
Drinking together	679.0	33.95	26.518814	156.0	19.500	15.378556	523.0	43.583333	27.971588
Drinking/eating alone	839.0	41.95	27.579839	355.0	44.375	30.360079	484.0	40.333333	25.430734
Eating together	281.0	14.05	11.796080	108.0	13.500	11.101802	173.0	14.416667	12.223327
Socializing	712.0	35.60	24.399590	241.0	30.125	27.903573	471.0	39.250000	20.972701
Attending a seminar	556.0	27.80	39.407613	35.0	4.375	5.998698	521.0	43.416667	44.210591
Meeting	836.0	41.80	44.361695	364.0	45.500	38.674927	472.0	39.333333	47.618857
Reading	666.0	33.30	37.361879	299.0	37.375	27.069067	367.0	30.583333	42.656493
TV	214.0	10.70	10.835590	106.0	13.250	10.461238	108.0	9.000000	10.747093
Cleaning and chores	436.0	21.80	13.948477	195.0	24.375	14.957753	241.0	20.083333	12.951566
Working	1226.0	61.30	66.131006	1004.0	125.500	58.350236	222.0	18.500000	20.962268
Cooking	128.0	6.40	4.442972	59.0	7.375	2.446298	69.0	5.750000	5.277705
Shopping	1069.0	53.45	51.334662	628.0	78.500	56.307193	441.0	36.750000	39.764148
Talking	211.0	10.55	9.630550	106.0	13.250	11.019868	105.0	8.750000	8.094494
Resting	639.0	31.95	26.481078	253.0	31.625	23.113511	386.0	32.166667	28.503898
Mobile	2315.0	115.75	96.759431	1564.0	195.500	101.471671	751.0	62.583333	40.901219
Plane	760.0	38.00	92.467832	48.0	6.000	7.826238	712.0	59.333333	114.332362

Table 20: User4 activities statistics

Day	All days			Routine			Non routine		
stat	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation
Public Transport	657.0	50.538462	33.258002	315.0	52.500000	29.736341	342.0	48.857143	35.918275
Driving	609.0	46.846154	46.787319	280.0	46.666667	32.045107	329.0	47.000000	56.437070
Walking outdoor	2196.0	168.923077	125.307291	620.0	103.333333	71.179741	1576.0	225.142857	134.051620
Walking indoor	1849.0	142.230769	115.061959	1088.0	181.333333	154.177027	761.0	108.714286	42.172145
Biking	18.0	1.384615	1.820332	7.0	1.166667	0.897527	11.0	1.571429	2.321154
Drinking together	611.0	47.000000	35.995726	142.0	23.666667	8.576454	469.0	67.000000	38.426182
Drinking/eating alone	855.0	65.769231	39.766924	196.0	32.666667	13.646326	659.0	94.142857	32.140000
Eating together	184.0	14.153846	14.021117	61.0	10.166667	7.819136	123.0	17.571429	16.952515
Socializing	612.0	47.076923	40.958631	311.0	51.833333	55.083018	301.0	43.000000	21.882805
Attending a seminar	959.0	73.769231	67.165873	222.0	37.000000	26.038433	737.0	105.285714	75.131721
Meeting	745.0	57.307692	39.485897	405.0	67.500000	37.446629	340.0	48.571429	39.092773
Reading	523.0	40.230769	19.434916	248.0	41.333333	15.975676	275.0	39.285714	21.926594
TV	336.0	25.846154	49.486356	272.0	45.333333	67.504732	64.0	9.142857	6.127889
Cleaning and chores	219.0	16.846154	17.168155	153.0	25.500000	21.815514	66.0	9.428571	4.499433
Working	1996.0	153.538462	155.545005	1488.0	248.000000	178.645832	508.0	72.571429	58.078413
Cooking	94.0	7.230769	5.963399	46.0	7.666667	6.920180	48.0	6.857143	4.969293
Shopping	1217.0	93.615385	60.321623	287.0	47.833333	23.525990	930.0	132.857143	54.283459
Talking	313.0	24.076923	11.357556	177.0	29.500000	11.615363	136.0	19.428571	8.780777
Resting	331.0	25.461538	19.746919	144.0	24.000000	15.405627	187.0	26.714286	22.745262
Mobile	1541.0	118.538462	41.485151	643.0	107.166667	36.393299	898.0	128.285714	43.070422
Plane	1181.0	90.846154	187.880832	880.0	146.666667	263.718705	301.0	43.000000	31.359665

Table 21: User5 activities statistics

Day	All days			Routine			Non routine		
stat	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation
Public Transport	167.0	10.4375	9.178567	137.0	9.785714	9.607873	30.0	15.0	2.0
Driving	185.0	11.5625	24.305269	79.0	5.642857	8.901101	106.0	53.0	47.0
Walking outdoor	1616.0	101.0000	83.915136	1062.0	75.857143	50.579297	554.0	277.0	55.0
Walking indoor	790.0	49.3750	18.827755	656.0	46.857143	18.066770	134.0	67.0	14.0
Biking	13.0	0.8125	1.379255	7.0	0.500000	0.906327	6.0	3.0	2.0
Drinking together	427.0	26.6875	30.571062	334.0	23.857143	29.364354	93.0	46.5	31.5
Drinking/eating alone	648.0	40.5000	26.322519	610.0	43.571429	26.762752	38.0	19.0	1.0
Eating together	266.0	16.6250	11.067266	259.0	18.500000	10.561047	7.0	3.5	1.5
Socializing	373.0	23.3125	23.655651	303.0	21.642857	24.659415	70.0	35.0	8.0
Attending a seminar	248.0	15.5000	9.137833	223.0	15.928571	9.602561	25.0	12.5	3.5
Meeting	907.0	56.6875	39.666924	856.0	61.142857	39.881713	51.0	25.5	18.5
Reading	300.0	18.7500	13.292385	268.0	19.142857	14.040030	32.0	16.0	5.0
TV	98.0	6.1250	3.705992	89.0	6.357143	3.902511	9.0	4.5	0.5
Cleaning and chores	446.0	27.8750	22.482980	371.0	26.500000	21.131392	75.0	37.5	28.5
Working	4581.0	286.3125	146.454651	4551.0	325.071429	111.765356	30.0	15.0	5.0
Cooking	122.0	7.6250	6.450533	113.0	8.071429	6.755572	9.0	4.5	1.5
Shopping	1075.0	67.1875	44.501712	829.0	59.214286	41.130765	246.0	123.0	21.0
Talking	613.0	38.3125	18.085625	581.0	41.500000	17.086962	32.0	16.0	2.0
Resting	246.0	15.3750	13.577900	171.0	12.214286	7.202961	75.0	37.5	23.5
Mobile	2198.0	137.3750	69.940041	2130.0	152.142857	62.012507	68.0	34.0	1.0
Plane	132.0	8.2500	18.053739	50.0	3.571429	7.752551	82.0	41.0	31.0

Table 22: User6 activities statistics

Day	All days			Routine			Non routine		
stat	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation	Total imgs.	Average	Standard deviation
Public Transport	369.0	28.384615	27.379213	84.0	12.000000	9.971388	285.0	47.500000	28.802488
Driving	184.0	14.153846	16.860724	94.0	13.428571	21.750909	90.0	15.000000	7.916228
Walking outdoor	1754.0	134.923077	66.544909	634.0	90.571429	37.833685	1120.0	186.666667	54.334356
Walking indoor	848.0	65.230769	40.472505	406.0	58.000000	13.959124	442.0	73.666667	56.476150
Biking	30.0	2.307692	4.102244	6.0	0.857143	0.989743	24.0	4.000000	5.477226
Drinking together	629.0	48.384615	47.701860	262.0	37.428571	45.559560	367.0	61.166667	46.958906
Drinking/eating alone	396.0	30.461538	20.927036	138.0	19.714286	10.235991	258.0	43.000000	23.122860
Eating together	283.0	21.769231	16.834734	141.0	20.142857	18.719192	142.0	23.666667	14.090974
Socializing	488.0	37.538462	32.641680	112.0	16.000000	7.671841	376.0	62.666667	32.668367
Attending a seminar	187.0	14.384615	17.265076	38.0	5.428571	4.655477	149.0	24.833333	20.440292
Meeting	517.0	39.769231	39.646814	352.0	50.285714	47.297582	165.0	27.500000	22.721136
Reading	871.0	67.000000	57.060022	411.0	58.714286	46.188169	460.0	76.666667	66.271328
TV	88.0	6.769231	5.264597	54.0	7.714286	4.130524	34.0	5.666667	6.155395
Cleaning and chores	97.0	7.461538	3.608012	54.0	7.714286	3.534091	43.0	7.166667	3.670453
Working	1720.0	132.307692	119.140760	1583.0	226.142857	77.460457	137.0	22.833333	38.697186
Cooking	54.0	4.153846	4.293934	27.0	3.857143	3.181580	27.0	4.500000	5.283622
Shopping	564.0	43.384615	27.736503	288.0	41.142857	16.199017	276.0	46.000000	36.715120
Talking	398.0	30.615385	24.081715	300.0	42.857143	25.914301	98.0	16.333333	9.706813
Resting	330.0	25.384615	11.868112	207.0	29.571429	11.757759	123.0	20.500000	9.979145
Mobile	925.0	71.153846	32.991482	582.0	83.142857	24.907994	343.0	57.166667	35.653035
Plane	225.0	17.307692	25.051899	16.0	2.285714	2.913725	209.0	34.833333	27.919030

Table 23: User7 activities statistics

8.3 Day logging data quality tables

Daily time logged in hours, and lifelogging sessions continuity in percents.

	Images	Time logged(h)	Logging completeness(%)
day 1	1383	6.668086	99.014641
day 2	1586	10.886792	47.351608
day 3	1926	11.997760	52.979989
day 4	1615	12.115335	87.342213
day 5	1643	14.807772	92.081771
day 6	1371	12.351454	51.512685
day 7	1379	12.400311	53.287560
day 8	1391	12.542583	86.930224
day 9	1425	12.849366	53.564444
day 10	1135	10.232822	45.649973
day 11	1363	12.257721	97.073771
day 12	1416	12.740364	94.044103
day 13	1493	13.170092	89.228268
day 14	1395	12.544428	52.284182
day 15	178	1.579233	20.981068

Table 24: User1 per day hours and completeness data

	Images	Time logged(h)	Logging completeness(%)
day 1	712	6.415018	44.927466
day 2	641	5.747584	65.412562
day 3	947	7.889510	60.599191
day 4	894	8.015313	73.361111
day 5	1013	9.079776	85.269459
day 6	1478	13.248146	85.221438
day 7	960	8.596941	43.699680
day 8	875	7.831262	91.940195
day 9	1060	9.486237	62.723528
day 10	1003	8.129524	54.976681

Table 25: User2 per day hours and completeness data

	Images	Time logged(h)	Logging completeness(%)
day 1	1627	10.069263	79.777606
day 2	1384	8.541632	84.577587
day 3	1490	7.217991	49.681218
day 4	648	3.101468	31.239431
day 5	1376	6.576232	49.909214
day 6	1851	10.298935	92.829656
day 7	1948	12.033751	85.764773
day 8	842	5.221279	40.002138
day 9	1397	8.637959	73.962166
day 10	1613	9.946010	74.061217
day 11	623	3.851236	31.084818
day 12	1204	7.434861	91.060790
day 13	1860	11.488190	92.736020
day 14	1206	7.447360	79.743304
day 15	1480	9.155794	78.551176
day 16	1057	6.520165	85.866970

Table 26: User3 per day hours and completeness data

	Images	Time logged(h)	Logging completeness(%)
day 1	1069	9.581738	94.136009
day 2	1010	9.052550	72.958670
day 3	834	7.008141	85.828574
day 4	1098	9.851457	90.648310
day 5	1190	10.683281	84.445393
day 6	938	8.422026	92.141906
day 7	904	8.102762	96.426379
day 8	1103	9.104749	62.265336
day 9	1151	9.726620	57.016041
day 10	858	7.289949	38.725974
day 11	935	8.386160	96.999667
day 12	973	8.243438	42.968761
day 13	1106	9.392951	84.433128
day 14	1069	8.979409	97.411101
day 15	1069	9.015210	59.836570
day 16	1014	8.628886	64.804398
day 17	782	6.613698	43.567703
day 18	874	7.727176	49.850070
day 19	1066	8.987779	57.449274
day 20	273	2.263773	13.537064
day 21	967	8.096323	90.698164

Table 27: User4 per day hours and completeness data

	Images	Time logged(h)	Logging completeness(%)
day 1	1302	11.668827	86.653279
day 2	728	6.594285	69.735695
day 3	1433	12.856291	79.363904
day 4	1179	10.559144	95.196511
day 5	1505	13.424368	55.960776
day 6	1444	12.898124	53.940090
day 7	1164	10.445368	46.884601
day 8	1270	11.345621	50.616824
day 9	1437	12.757669	54.893338
day 10	1752	14.778385	86.348960
day 11	1600	13.479236	98.424507
day 12	1170	10.498035	94.404434
day 13	1062	8.900443	97.228328

Table 28: User5 per day hours and completeness data

	Images	Time logged(h)	Logging completeness(%)
day 1	742	4.738254	82.692044
day 2	728	4.554624	88.644895
day 3	1435	9.118927	84.138036
day 4	1318	8.210832	88.637985
day 5	969	5.962689	61.255261
day 6	1267	7.889065	61.681508
day 7	1678	10.389478	75.850986
day 8	1078	6.677163	78.298984
day 9	882	5.477076	39.271577
day 10	890	5.527800	82.378113
day 11	815	7.308692	61.701314
day 12	92	0.820487	72.896173
day 13	837	7.475371	70.513131
day 14	1049	9.368637	39.042767
day 15	955	8.569821	35.722472
day 16	497	4.455833	100.000000
day 17	622	5.577782	87.092361
day 18	738	6.603217	29.897223

Table 29: User6 per day hours and completeness data

	Images	Time logged(h)	Logging completeness(%)
day 1	781	6.566493	93.691791
day 2	743	6.665381	61.435228
day 3	747	6.718038	61.675813
day 4	844	7.596333	70.729353
day 5	809	7.259559	78.160158
day 6	688	5.788798	89.919203
day 7	760	6.824401	84.734233
day 8	850	7.623879	65.933083
day 9	962	8.633347	93.120950
day 10	718	6.454849	71.495464
day 11	1316	11.817222	100.000000
day 12	792	7.111340	97.888677
day 13	947	8.501618	95.658144

Table 30: User7 per day hours and completeness data

8.4 Full results tables

epsilon	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
0.45 avg	man	all	67.65	42.86	40.00	75.00	85.00	53.85	87.50	69.23
		routine	50.00	27.27	33.33	66.67	62.50	0.00	83.33	42.86
		n.routine	95.00	100.00	50.00	100.00	100.00	100.00	100.00	100.00
	euc	all	57.84	57.14	40.00	56.25	80.00	30.77	75.00	46.15
		routine	35.48	45.45	0.00	41.67	50.00	0.00	66.67	0.00
		n.routine	92.50	100.00	100.00	100.00	100.00	57.14	100.00	100.00
	cos	all	70.59	64.29	40.00	62.50	80.00	76.92	87.50	69.23
		routine	51.61	54.55	0.00	50.00	50.00	50.00	83.33	42.86
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
0.51 avg	man	all	60.78	21.43	40.00	62.50	85.00	53.85	75.00	69.23
		routine	35.48	0.00	0.00	50.00	62.50	0.00	66.67	42.86
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	euc	all	62.75	71.43	40.00	68.75	70.00	23.08	81.25	69.23
		routine	56.45	72.73	0.00	58.33	75.00	0.00	91.67	42.86
		n.routine	72.50	66.67	100.00	100.00	66.67	42.86	50.00	100.00
	cos	all	78.43	85.71	40.00	75.00	85.00	76.92	87.50	84.62
		routine	66.13	81.82	0.00	66.67	62.50	50.00	91.67	71.43
		n.routine	97.50	100.00	100.00	100.00	100.00	100.00	75.00	100.00
0.56 avg	man	all	66.67	42.86	50.00	75.00	90.00	30.77	87.50	69.23
		routine	53.23	27.27	50.00	66.67	75.00	0.00	83.33	42.86
		n.routine	87.50	100.00	50.00	100.00	100.00	57.14	100.00	100.00
	euc	all	66.67	71.43	50.00	75.00	60.00	46.15	87.50	69.23
		routine	70.97	72.73	33.33	66.67	100.00	50.00	100.00	42.86
		n.routine	60.00	66.67	75.00	100.00	33.33	42.86	50.00	100.00
	cos	all	75.49	92.86	50.00	75.00	80.00	46.15	87.50	84.62
		routine	75.81	100.00	33.33	66.67	87.50	50.00	91.67	71.43
		n.routine	75.00	66.67	75.00	100.00	75.00	42.86	75.00	100.00

Table 31: DBSCAN for histograms, 3 values of epsilon are used: 45% of average dist, 51% of average dist and 56% of avg dist.

distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
man	all	58.82	64.29	40.00	87.50	50.00	53.85	62.50	46.15
	routine	77.42	72.73	33.33	91.67	100.00	83.33	75.00	71.43
	n.routine	30.00	33.33	50.00	75.00	16.67	28.57	25.00	16.67
euc	all	51.96	57.14	40.00	56.25	40.00	69.23	56.25	46.15
	routine	70.97	63.64	33.33	75.00	100.00	83.33	66.67	71.43
	n.routine	22.50	33.33	50.00	0.00	0.00	57.14	25.00	16.67
cos	all	62.75	64.29	40.00	75.00	50.00	69.23	81.25	53.85
	routine	80.65	72.73	33.33	91.67	100.00	83.33	91.67	71.43
	n.routine	35.00	33.33	50.00	25.00	16.67	57.14	50.00	33.33

Table 32: Spectral clustering for histograms.

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	75.49	92.86	60.00	100.0	65.00	38.46	87.5	76.92
		routine	95.16	100.00	66.67	100.0	100.00	83.33	100.0	100.00
		n.routine	45.00	66.67	50.00	100.0	41.67	0.00	50.0	50.00
	euc	all	71.57	92.86	50.00	100.0	55.00	38.46	87.5	69.23
		routine	93.55	100.00	50.00	100.0	100.00	83.33	100.0	100.00
		n.routine	37.50	66.67	50.00	100.0	25.00	0.00	50.0	33.33
	cos	all	72.55	92.86	50.00	100.0	55.00	38.46	87.5	76.92
		routine	95.16	100.00	66.67	100.0	100.00	83.33	100.0	100.00
		n.routine	37.50	66.67	25.00	100.0	25.00	0.00	50.0	50.00
3	man	all	73.53	92.86	50.00	100.0	60.00	38.46	87.5	76.92
		routine	93.55	100.00	50.00	100.0	100.00	83.33	100.0	100.00
		n.routine	42.50	66.67	50.00	100.0	33.33	0.00	50.0	50.00
	euc	all	68.63	85.71	40.00	100.0	50.00	38.46	87.5	69.23
		routine	91.94	90.91	50.00	100.0	100.00	83.33	100.0	100.00
		n.routine	32.50	66.67	25.00	100.0	16.67	0.00	50.0	33.33
	cos	all	73.53	92.86	50.00	100.0	55.00	38.46	87.5	84.62
		routine	95.16	100.00	66.67	100.0	100.00	83.33	100.0	100.00
		n.routine	40.00	66.67	25.00	100.0	25.00	0.00	50.0	66.67
4	man	all	72.55	85.71	50.00	100.0	60.00	46.15	87.5	69.23
		routine	91.94	90.91	50.00	100.0	100.00	83.33	100.0	100.00
		n.routine	42.50	66.67	50.00	100.0	33.33	14.29	50.0	33.33
	euc	all	68.63	85.71	40.00	100.0	50.00	38.46	87.5	69.23
		routine	91.94	90.91	50.00	100.0	100.00	83.33	100.0	100.00
		n.routine	32.50	66.67	25.00	100.0	16.67	0.00	50.0	33.33
	cos	all	72.55	92.86	40.00	100.0	55.00	38.46	87.5	84.62
		routine	91.94	100.00	50.00	100.0	100.00	83.33	100.0	85.71
		n.routine	42.50	66.67	25.00	100.0	25.00	0.00	50.0	83.33

Table 33: Local novelty detection for histograms, 3 values of k are used: k=2, k=3, k=4

epsilon	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
0.8 avg	man	all	47.06	85.71	50.00	43.75	30.00	23.08	75.00	23.08
		routine	72.58	90.91	83.33	58.33	75.00	50.00	100.00	28.57
		n.routine	7.50	66.67	0.00	0.00	0.00	0.00	0.00	16.67
	eucl	all	44.12	21.43	40.00	12.50	70.00	23.08	81.25	46.15
		routine	27.42	0.00	0.00	8.33	37.50	16.67	100.00	0.00
		n.routine	70.00	100.00	100.00	25.00	91.67	28.57	25.00	100.00
	cos	all	39.22	21.43	40.00	25.00	60.00	53.85	25.00	46.15
		routine	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
0.87 avg	man	all	56.86	85.71	50.00	62.50	35.00	46.15	75.00	46.15
		routine	88.71	90.91	83.33	83.33	87.50	100.00	100.00	71.43
		n.routine	7.50	66.67	0.00	0.00	0.00	0.00	0.00	16.67
	eucl	all	47.06	78.57	40.00	25.00	50.00	15.38	75.00	38.46
		routine	58.06	72.73	66.67	33.33	62.50	33.33	100.00	14.29
		n.routine	30.00	100.00	0.00	0.00	41.67	0.00	0.00	66.67
	cos	all	42.16	21.43	40.00	25.00	60.00	53.85	43.75	46.15
		routine	6.45	0.00	0.00	0.00	0.00	0.00	33.33	0.00
		n.routine	97.50	100.00	100.00	100.00	100.00	100.00	75.00	100.00
0.95 avg	man	all	60.78	78.57	50.00	75.00	40.00	46.15	75.00	61.54
		routine	96.77	90.91	83.33	100.00	100.00	100.00	100.00	100.00
		n.routine	5.00	33.33	0.00	0.00	0.00	0.00	0.00	16.67
	eucl	all	57.84	78.57	50.00	75.00	35.00	46.15	75.00	46.15
		routine	91.94	90.91	83.33	100.00	87.50	100.00	100.00	71.43
		n.routine	5.00	33.33	0.00	0.00	0.00	0.00	0.00	16.67
	cos	all	68.63	71.43	40.00	75.00	70.00	61.54	87.50	61.54
		routine	77.42	63.64	66.67	66.67	100.00	66.67	100.00	71.43
		n.routine	55.00	100.00	0.00	100.00	50.00	57.14	50.00	50.00

Table 34: DBSCAN for extended matrices, optimistic approach, 3 values of epsilon are used: 80% of average dist, 87% of average dist and 95% of avg dist.

epsilon	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
0.8 avg	man	all	46.08	42.86	30.00	50.00	45.00	53.85	68.75	23.08
		routine	40.32	36.36	33.33	58.33	0.00	50.00	75.00	0.00
		n.routine	55.00	66.67	25.00	25.00	75.00	57.14	50.00	50.00
	eucl	all	44.12	50.00	40.00	18.75	55.00	53.85	50.00	38.46
		routine	25.81	36.36	0.00	16.67	37.50	33.33	33.33	14.29
		n.routine	72.50	100.00	100.00	25.00	66.67	71.43	100.00	66.67
0.87 avg	man	all	57.84	78.57	40.00	43.75	55.00	53.85	87.50	38.46
		routine	70.97	81.82	50.00	58.33	62.50	83.33	100.00	42.86
		n.routine	37.50	66.67	25.00	0.00	50.00	28.57	50.00	33.33
	eucl	all	53.92	71.43	40.00	43.75	50.00	46.15	75.00	46.15
		routine	67.74	90.91	50.00	50.00	62.50	66.67	75.00	71.43
		n.routine	32.50	0.00	25.00	25.00	41.67	28.57	75.00	16.67
0.95 avg	man	all	60.78	85.71	50.00	56.25	45.00	38.46	87.50	61.54
		routine	90.32	100.00	83.33	75.00	87.50	83.33	100.00	100.00
		n.routine	15.00	33.33	0.00	0.00	16.67	0.00	50.00	16.67
	eucl	all	59.80	78.57	50.00	75.00	45.00	30.77	81.25	53.85
		routine	93.55	100.00	83.33	91.67	100.00	66.67	100.00	100.00
		n.routine	7.50	0.00	0.00	25.00	8.33	0.00	25.00	0.00

Table 35: DBSCAN for extended matrices, pessimistic approach, 3 values of epsilon are used: 80% of average dist, 87% of average dist and 95% of avg dist.

distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
man	all	57.84	64.29	40.00	43.75	65.00	61.54	68.75	53.85
	routine	66.13	63.64	33.33	58.33	100.00	66.67	66.67	71.43
	n.routine	45.00	66.67	50.00	0.00	41.67	57.14	75.00	33.33
eucl	all	69.61	85.71	60.00	56.25	65.00	69.23	75.00	76.92
	routine	79.03	81.82	50.00	66.67	100.00	83.33	75.00	100.00
	n.routine	55.00	100.00	75.00	25.00	41.67	57.14	75.00	50.00
cos	all	56.86	64.29	20.00	81.25	45.00	61.54	62.50	53.85
	routine	77.42	72.73	16.67	91.67	100.00	83.33	83.33	71.43
	n.routine	25.00	33.33	25.00	50.00	8.33	42.86	0.00	33.33

Table 36: Spectral clustering for extended matrices, optimistic approach.

distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
man	all	50.98	71.43	30.00	37.50	50.00	38.46	75.00	46.15
	routine	64.52	72.73	16.67	50.00	100.00	50.00	83.33	57.14
	n.routine	30.00	66.67	50.00	0.00	16.67	28.57	50.00	33.33
eucl	all	57.84	85.71	40.00	75.00	45.00	38.46	81.25	30.77
	routine	80.65	90.91	33.33	91.67	100.00	66.67	100.00	42.86
	n.routine	22.50	66.67	50.00	25.00	8.33	14.29	25.00	16.67

Table 37: Spectral clustering for extended matrices, pessimistic approach

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	41.18	85.71	50.00	12.50	40.00	15.38	50.00	38.46
		routine	50.00	90.91	66.67	16.67	50.00	33.33	41.67	57.14
		n.routine	27.50	66.67	25.00	0.00	33.33	0.00	75.00	16.67
	eucl	all	39.22	78.57	40.00	12.50	45.00	23.08	37.50	38.46
		routine	41.94	81.82	50.00	16.67	50.00	16.67	25.00	57.14
		n.routine	35.00	66.67	25.00	0.00	41.67	28.57	75.00	16.67
	cos	all	70.59	71.43	50.00	81.25	70.00	69.23	81.25	61.54
		routine	72.58	63.64	50.00	83.33	62.50	66.67	91.67	71.43
		n.routine	67.50	100.00	50.00	75.00	75.00	71.43	50.00	50.00
3	man	all	41.18	85.71	50.00	12.50	45.00	15.38	43.75	38.46
		routine	46.77	90.91	50.00	16.67	50.00	33.33	33.33	57.14
		n.routine	32.50	66.67	50.00	0.00	41.67	0.00	75.00	16.67
	eucl	all	40.20	78.57	60.00	12.50	45.00	7.69	43.75	38.46
		routine	43.55	81.82	50.00	16.67	50.00	16.67	33.33	57.14
		n.routine	35.00	66.67	75.00	0.00	41.67	0.00	75.00	16.67
	cos	all	73.53	78.57	40.00	87.50	70.00	69.23	81.25	76.92
		routine	72.58	72.73	33.33	83.33	62.50	66.67	91.67	71.43
		n.routine	75.00	100.00	50.00	100.00	75.00	71.43	50.00	83.33
4	man	all	44.12	85.71	60.00	18.75	45.00	15.38	50.00	38.46
		routine	50.00	90.91	50.00	25.00	50.00	33.33	41.67	57.14
		n.routine	35.00	66.67	75.00	0.00	41.67	0.00	75.00	16.67
	eucl	all	40.20	71.43	60.00	18.75	45.00	7.69	43.75	38.46
		routine	45.16	81.82	50.00	25.00	50.00	16.67	33.33	57.14
		n.routine	32.50	33.33	75.00	0.00	41.67	0.00	75.00	16.67
	cos	all	73.53	92.86	20.00	93.75	70.00	69.23	81.25	69.23
		routine	75.81	90.91	33.33	91.67	62.50	66.67	91.67	57.14
		n.routine	70.00	100.00	0.00	100.00	75.00	71.43	50.00	83.33

Table 38: Local novelty detection for extended matrices, optimistic approach, 3 values of k are used: k=2, k=3,k=4

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	54.90	71.43	40.00	31.25	55.00	53.85	68.75	61.54
		routine	64.52	72.73	66.67	33.33	62.50	50.00	75.00	100.00
		n.routine	40.00	66.67	0.00	25.00	50.00	57.14	50.00	16.67
	eucl	all	52.94	57.14	40.00	43.75	60.00	30.77	87.50	38.46
		routine	62.90	54.55	66.67	50.00	62.50	50.00	91.67	57.14
		n.routine	37.50	66.67	0.00	25.00	58.33	14.29	75.00	16.67
3	man	all	58.82	71.43	30.00	37.50	70.00	53.85	75.00	61.54
		routine	67.74	72.73	50.00	41.67	75.00	50.00	83.33	100.00
		n.routine	45.00	66.67	0.00	25.00	66.67	57.14	50.00	16.67
	eucl	all	52.94	64.29	40.00	43.75	60.00	23.08	81.25	46.15
		routine	64.52	63.64	66.67	50.00	62.50	50.00	83.33	71.43
		n.routine	35.00	66.67	0.00	25.00	58.33	0.00	75.00	16.67
4	man	all	57.84	64.29	30.00	31.25	70.00	61.54	75.00	61.54
		routine	64.52	63.64	50.00	33.33	75.00	50.00	83.33	100.00
		n.routine	47.50	66.67	0.00	25.00	66.67	71.43	50.00	16.67
	eucl	all	53.92	64.29	40.00	43.75	60.00	23.08	81.25	53.85
		routine	66.13	63.64	66.67	50.00	62.50	50.00	83.33	85.71
		n.routine	35.00	66.67	0.00	25.00	58.33	0.00	75.00	16.67

Table 39: Local novelty detection for extended matrices, pessimistic approach, 3 values of k are used: k=2,k=3,k=4

epsilon	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
0.77 avg	man	all	39.22	21.43	40.00	25.00	60.00	53.85	25.00	46.15
		routine	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	eucl	all	55.88	57.14	50.00	81.25	45.00	61.54	50.00	46.15
		routine	69.35	63.64	50.00	75.00	75.00	66.67	66.67	85.71
		n.routine	35.00	33.33	50.00	100.00	25.00	57.14	0.00	0.00
	cos	all	39.22	21.43	40.00	25.00	60.00	53.85	25.00	46.15
		routine	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
0.85 avg	man	all	38.24	21.43	40.00	25.00	40.00	76.92	25.00	46.15
		routine	4.84	0.00	0.00	0.00	0.00	50.00	0.00	0.00
		n.routine	90.00	100.00	100.00	100.00	66.67	100.00	100.00	100.00
	eucl	all	54.90	57.14	40.00	75.00	40.00	46.15	68.75	53.85
		routine	80.65	63.64	66.67	83.33	87.50	66.67	91.67	100.00
		n.routine	15.00	33.33	0.00	50.00	8.33	28.57	0.00	0.00
	cos	all	42.16	21.43	40.00	25.00	60.00	76.92	25.00	46.15
		routine	4.84	0.00	0.00	0.00	0.00	50.00	0.00	0.00
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
0.93 avg	man	all	59.80	71.43	40.00	43.75	50.00	23.08	87.50	100.00
		routine	67.74	63.64	0.00	41.67	100.00	50.00	100.00	100.00
		n.routine	47.50	100.00	100.00	50.00	16.67	0.00	50.00	100.00
	eucl	all	59.80	78.57	50.00	75.00	40.00	53.85	68.75	53.85
		routine	93.55	90.91	83.33	91.67	100.00	100.00	91.67	100.00
		n.routine	7.50	33.33	0.00	25.00	0.00	14.29	0.00	0.00
	cos	all	64.71	71.43	40.00	25.00	65.00	76.92	87.50	84.62
		routine	58.06	63.64	0.00	16.67	87.50	50.00	100.00	71.43
		n.routine	75.00	100.00	100.00	50.00	50.00	100.00	50.00	100.00

Table 40: DBSCAN for blocks matrices, optimistic approach, 3 values of epsilon are used: 77% of average dist, 85% of average dist and 93% of avg dist.

epsilon	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
0.77 avg	man	all	42.16	21.43	40.00	25.00	60.00	76.92	25.00	46.15
		routine	4.84	0.00	0.00	0.00	0.00	50.00	0.00	0.00
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	eucl	all	54.90	50.00	60.00	50.00	50.00	61.54	62.50	53.85
		routine	62.90	36.36	50.00	50.00	62.50	66.67	83.33	100.00
		n.routine	42.50	100.00	75.00	50.00	41.67	57.14	0.00	0.00
0.85 avg	man	all	45.10	50.00	40.00	18.75	50.00	84.62	31.25	46.15
		routine	22.58	36.36	0.00	16.67	12.50	66.67	25.00	0.00
		n.routine	80.00	100.00	100.00	25.00	75.00	100.00	50.00	100.00
	eucl	all	54.90	57.14	30.00	68.75	40.00	61.54	68.75	53.85
		routine	77.42	63.64	50.00	83.33	75.00	66.67	91.67	100.00
		n.routine	20.00	33.33	0.00	25.00	16.67	57.14	0.00	0.00
0.93 avg	man	all	60.78	78.57	50.00	68.75	40.00	46.15	87.50	53.85
		routine	91.94	81.82	83.33	91.67	87.50	100.00	100.00	100.00
		n.routine	12.50	66.67	0.00	0.00	8.33	0.00	50.00	0.00
	eucl	all	57.84	71.43	40.00	75.00	40.00	46.15	75.00	53.85
		routine	90.32	81.82	66.67	91.67	100.00	83.33	100.00	100.00
		n.routine	7.50	33.33	0.00	25.00	0.00	14.29	0.00	0.00

Table 41: DBSCAN for blocks matrices, pessimistic approach, 3 values of epsilon are used: 77% of average dist, 85% of average dist and 93% of avg dist.

distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
man	all	59.80	92.86	20.00	75.00	50.00	46.15	68.75	53.85
	routine	77.42	90.91	16.67	83.33	100.00	66.67	83.33	71.43
	n.routine	32.50	100.00	25.00	50.00	16.67	28.57	25.00	33.33
eucl	all	61.76	64.29	60.00	75.00	50.00	53.85	75.00	53.85
	routine	80.65	72.73	50.00	83.33	100.00	83.33	91.67	71.43
	n.routine	32.50	33.33	75.00	50.00	16.67	28.57	25.00	33.33
cos	all	52.94	85.71	20.00	62.50	40.00	30.77	68.75	53.85
	routine	70.97	81.82	16.67	75.00	87.50	50.00	83.33	71.43
	n.routine	25.00	100.00	25.00	25.00	8.33	14.29	25.00	33.33

Table 42: Spectral clustering for blocks matrices, optimistic approach.

distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
man	all	56.86	85.71	40.00	68.75	50.00	30.77	75.00	38.46
	routine	74.19	81.82	33.33	75.00	100.00	50.00	91.67	57.14
	n.routine	30.00	100.00	50.00	50.00	16.67	14.29	25.00	16.67
eucl	all	62.75	64.29	20.00	93.75	50.00	61.54	75.00	61.54
	routine	79.03	72.73	16.67	100.00	87.50	83.33	83.33	85.71
	n.routine	37.50	33.33	25.00	75.00	25.00	42.86	50.00	33.33

Table 43: Spectral clustering for blocks matrices, pessimistic approach.

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	65.69	64.29	30.00	37.50	70.00	76.92	87.50	84.62
		routine	64.52	63.64	33.33	33.33	87.50	50.00	100.00	71.43
		n.routine	67.50	66.67	25.00	50.00	58.33	100.00	50.00	100.00
	eucl	all	52.94	57.14	40.00	75.00	45.00	61.54	56.25	30.77
		routine	53.23	63.64	50.00	66.67	37.50	66.67	58.33	14.29
		n.routine	52.50	33.33	25.00	100.00	50.00	57.14	50.00	50.00
	cos	all	67.65	78.57	50.00	43.75	55.00	76.92	87.50	84.62
		routine	70.97	72.73	66.67	41.67	87.50	50.00	100.00	71.43
		n.routine	62.50	100.00	25.00	50.00	33.33	100.00	50.00	100.00
3	man	all	67.65	78.57	30.00	50.00	65.00	69.23	87.50	84.62
		routine	69.35	72.73	33.33	50.00	87.50	50.00	100.00	71.43
		n.routine	65.00	100.00	25.00	50.00	50.00	85.71	50.00	100.00
	eucl	all	52.94	57.14	40.00	68.75	50.00	61.54	56.25	30.77
		routine	50.00	63.64	50.00	66.67	37.50	50.00	50.00	14.29
		n.routine	57.50	33.33	25.00	75.00	58.33	71.43	75.00	50.00
	cos	all	69.61	78.57	60.00	43.75	60.00	76.92	87.50	84.62
		routine	70.97	72.73	66.67	41.67	87.50	50.00	100.00	71.43
		n.routine	67.50	100.00	50.00	50.00	41.67	100.00	50.00	100.00
4	man	all	64.71	71.43	50.00	50.00	60.00	46.15	87.50	84.62
		routine	69.35	72.73	33.33	50.00	87.50	50.00	100.00	71.43
		n.routine	57.50	66.67	75.00	50.00	41.67	42.86	50.00	100.00
	eucl	all	53.92	57.14	40.00	75.00	50.00	61.54	56.25	30.77
		routine	53.23	63.64	50.00	75.00	37.50	50.00	58.33	14.29
		n.routine	55.00	33.33	25.00	75.00	58.33	71.43	50.00	50.00
	cos	all	68.63	78.57	70.00	43.75	60.00	61.54	87.50	84.62
		routine	70.97	72.73	66.67	41.67	87.50	50.00	100.00	71.43
		n.routine	65.00	100.00	75.00	50.00	41.67	71.43	50.00	100.00

Table 44: Local novelty detection for blocks matrices, optimistic approach, k=2, k=3, k=4.

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	66.67	71.43	50.0	43.75	65.00	76.92	87.50	69.23
		routine	67.74	63.64	50.0	50.00	87.50	50.00	100.00	57.14
		n.routine	65.00	100.00	50.0	25.00	50.00	100.00	50.00	83.33
	eucl	all	50.00	50.00	60.0	68.75	50.00	61.54	31.25	30.77
		routine	46.77	54.55	50.0	75.00	37.50	66.67	25.00	14.29
		n.routine	55.00	33.33	75.0	50.00	58.33	57.14	50.00	50.00
3	man	all	64.71	71.43	50.0	37.50	60.00	69.23	87.50	76.92
		routine	64.52	63.64	50.0	33.33	87.50	33.33	100.00	71.43
		n.routine	65.00	100.00	50.0	50.00	41.67	100.00	50.00	83.33
	eucl	all	50.00	50.00	50.0	68.75	55.00	61.54	31.25	30.77
		routine	48.39	54.55	50.0	75.00	50.00	66.67	25.00	14.29
		n.routine	52.50	33.33	50.0	50.00	58.33	57.14	50.00	50.00
4	man	all	62.75	71.43	50.0	25.00	65.00	61.54	87.50	76.92
		routine	59.68	63.64	50.0	16.67	87.50	16.67	100.00	71.43
		n.routine	67.50	100.00	50.0	50.00	50.00	100.00	50.00	83.33
	eucl	all	50.98	57.14	50.0	68.75	50.00	69.23	31.25	30.77
		routine	50.00	63.64	50.0	75.00	50.00	66.67	25.00	14.29
		n.routine	52.50	33.33	50.0	50.00	50.00	71.43	50.00	50.00

Table 45: Local novelty detection for blocks matrices, pessimistic approach, k=2, k=3, k=4.

epsilon	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
0.78 avg	man	all	60.78	42.86	30.00	50.00	75.00	30.77	87.50	92.31
		routine	59.68	27.27	33.33	41.67	75.00	50.00	100.00	85.71
		n.routine	62.50	100.00	25.00	75.00	75.00	14.29	50.00	100.00
	eucl	all	51.96	21.43	40.00	25.00	80.00	53.85	81.25	46.15
		routine	24.19	0.00	0.00	0.00	50.00	0.00	91.67	0.00
		n.routine	95.00	100.00	100.00	100.00	100.00	100.00	50.00	100.00
	cos	all	46.08	21.43	40.00	68.75	65.00	23.08	43.75	46.15
		routine	27.42	0.00	0.00	58.33	50.00	0.00	50.00	0.00
		n.routine	75.00	100.00	100.00	100.00	75.00	42.86	25.00	100.00
0.84 avg	man	all	59.80	64.29	40.00	62.50	60.00	38.46	75.00	69.23
		routine	77.42	54.55	50.00	75.00	87.50	83.33	100.00	85.71
		n.routine	32.50	100.00	25.00	25.00	41.67	0.00	0.00	50.00
	eucl	all	64.71	42.86	40.00	62.50	75.00	38.46	87.50	92.31
		routine	54.84	27.27	0.00	50.00	50.00	50.00	100.00	85.71
		n.routine	80.00	100.00	100.00	100.00	91.67	28.57	50.00	100.00
	cos	all	61.76	21.43	40.00	81.25	70.00	38.46	75.00	92.31
		routine	61.29	0.00	0.00	75.00	87.50	66.67	100.00	85.71
		n.routine	62.50	100.00	100.00	100.00	58.33	14.29	0.00	100.00
0.87 avg	man	all	61.76	78.57	50.00	62.50	55.00	38.46	75.00	69.23
		routine	85.48	72.73	66.67	83.33	100.00	83.33	100.00	85.71
		n.routine	25.00	100.00	25.00	0.00	25.00	0.00	0.00	50.00
	eucl	all	64.71	42.86	60.00	75.00	70.00	30.77	75.00	92.31
		routine	67.74	27.27	50.00	75.00	75.00	50.00	100.00	85.71
		n.routine	60.00	100.00	75.00	75.00	66.67	14.29	0.00	100.00
	cos	all	66.67	42.86	40.00	93.75	70.00	38.46	75.00	92.31
		routine	69.35	27.27	0.00	91.67	87.50	66.67	100.00	85.71
		n.routine	62.50	100.00	100.00	100.00	58.33	14.29	0.00	100.00

Table 46: DBSCAN for fixed size matrices, eps= 78%, 84% and 87% of average distance.

distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
man	all	61.76	64.29	60.00	56.25	50.00	46.15	87.50	69.23
	routine	79.03	72.73	50.00	75.00	100.00	66.67	91.67	85.71
	n.routine	35.00	33.33	75.00	0.00	16.67	28.57	75.00	50.00
eucl	all	66.67	78.57	40.00	75.00	50.00	53.85	87.50	76.92
	routine	87.10	81.82	33.33	91.67	100.00	83.33	100.00	100.00
	n.routine	35.00	66.67	50.00	25.00	16.67	28.57	50.00	50.00
cos	all	61.76	42.86	60.00	87.50	50.00	46.15	81.25	61.54
	routine	82.26	54.55	50.00	100.00	100.00	66.67	100.00	85.71
	n.routine	30.00	0.00	75.00	50.00	16.67	28.57	25.00	33.33

Table 47: Spectral clustering for fixed size matrices.

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	67.65	92.86	50.00	62.50	75.00	30.77	81.25	69.23
		routine	80.65	100.00	83.33	75.00	62.50	50.00	83.33	100.00
		n.routine	47.50	66.67	0.00	25.00	83.33	14.29	75.00	33.33
	eucl	all	72.55	92.86	50.00	75.00	80.00	46.15	75.00	76.92
		routine	77.42	100.00	83.33	75.00	62.50	50.00	66.67	100.00
		n.routine	65.00	66.67	0.00	75.00	91.67	42.86	100.00	50.00
	cos	all	68.63	64.29	40.00	100.00	70.00	38.46	62.50	92.31
		routine	74.19	81.82	33.33	100.00	87.50	66.67	50.00	85.71
		n.routine	60.00	0.00	50.00	100.00	58.33	14.29	100.00	100.00
3	man	all	65.69	92.86	40.00	62.50	70.00	30.77	87.50	61.54
		routine	79.03	100.00	66.67	75.00	62.50	50.00	83.33	100.00
		n.routine	45.00	66.67	0.00	25.00	75.00	14.29	100.00	16.67
	eucl	all	71.57	92.86	50.00	75.00	75.00	46.15	75.00	76.92
		routine	75.81	100.00	66.67	75.00	62.50	50.00	66.67	100.00
		n.routine	65.00	66.67	25.00	75.00	83.33	42.86	100.00	50.00
	cos	all	66.67	64.29	60.00	93.75	70.00	38.46	43.75	92.31
		routine	70.97	81.82	33.33	91.67	87.50	66.67	41.67	85.71
		n.routine	60.00	0.00	100.00	100.00	58.33	14.29	50.00	100.00
4	man	all	67.65	85.71	50.00	68.75	70.00	30.77	93.75	61.54
		routine	80.65	90.91	66.67	75.00	75.00	50.00	91.67	100.00
		n.routine	47.50	66.67	25.00	50.00	66.67	14.29	100.00	16.67
	eucl	all	71.57	92.86	50.00	75.00	70.00	46.15	81.25	76.92
		routine	79.03	100.00	66.67	75.00	75.00	50.00	75.00	100.00
		n.routine	60.00	66.67	25.00	75.00	66.67	42.86	100.00	50.00
	cos	all	65.69	64.29	50.00	93.75	70.00	38.46	43.75	92.31
		routine	70.97	81.82	33.33	91.67	87.50	66.67	41.67	85.71
		n.routine	57.50	0.00	75.00	100.00	58.33	14.29	50.00	100.00

Table 48: Local novelty detection for fixed size matrices, k=2, k=3 and k=4

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	74.51	71.43	60.00	100.0	75.00	30.77	81.25	92.31
		routine	82.26	63.64	83.33	100.0	75.00	50.00	91.67	100.00
		n.routine	62.50	100.00	25.00	100.0	75.00	14.29	50.00	83.33
	euc	all	70.59	71.43	60.00	100.0	55.00	30.77	87.50	84.62
		routine	83.87	63.64	83.33	100.0	75.00	50.00	100.00	100.00
		n.routine	50.00	100.00	25.00	100.0	41.67	14.29	50.00	66.67
	cos	all	75.49	71.43	60.00	100.0	75.00	30.77	87.50	92.31
		routine	83.87	63.64	83.33	100.0	75.00	50.00	100.00	100.00
		n.routine	62.50	100.00	25.00	100.0	75.00	14.29	50.00	83.33
3	man	all	76.47	85.71	60.00	100.0	75.00	30.77	81.25	92.31
		routine	85.48	81.82	83.33	100.0	75.00	50.00	91.67	100.00
		n.routine	62.50	100.00	25.00	100.0	75.00	14.29	50.00	83.33
	euc	all	71.57	78.57	60.00	100.0	50.00	38.46	87.50	84.62
		routine	85.48	72.73	83.33	100.0	75.00	50.00	100.00	100.00
		n.routine	50.00	100.00	25.00	100.0	33.33	28.57	50.00	66.67
	cos	all	76.47	78.57	60.00	100.0	70.00	38.46	87.50	92.31
		routine	85.48	72.73	83.33	100.0	75.00	50.00	100.00	100.00
		n.routine	62.50	100.00	25.00	100.0	66.67	28.57	50.00	83.33
4	man	all	75.49	85.71	60.00	100.0	70.00	30.77	87.50	84.62
		routine	87.10	81.82	83.33	100.0	75.00	50.00	100.00	100.00
		n.routine	57.50	100.00	25.00	100.0	66.67	14.29	50.00	66.67
	euc	all	72.55	85.71	60.00	100.0	50.00	38.46	87.50	84.62
		routine	87.10	81.82	83.33	100.0	75.00	50.00	100.00	100.00
		n.routine	50.00	100.00	25.00	100.0	33.33	28.57	50.00	66.67
	cos	all	74.51	85.71	60.00	100.0	60.00	38.46	87.50	84.62
		routine	87.10	81.82	83.33	100.0	75.00	50.00	100.00	100.00
		n.routine	55.00	100.00	25.00	100.0	50.00	28.57	50.00	66.67

Table 49: Feature Vector. Local novelty detection for Histogram, k=2, k=3 and k=4

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	80.39	78.57	80.00	93.75	85.00	46.15	81.25	92.31
		routine	85.48	72.73	83.33	100.00	87.50	50.00	91.67	100.00
		n.routine	72.50	100.00	75.00	75.00	83.33	42.86	50.00	83.33
	eucl	all	62.75	57.14	50.00	81.25	60.00	61.54	68.75	53.85
		routine	67.74	63.64	66.67	75.00	37.50	66.67	83.33	71.43
		n.routine	55.00	33.33	25.00	100.00	75.00	57.14	25.00	33.33
	cos	all	77.45	78.57	60.00	87.50	85.00	46.15	87.50	84.62
		routine	83.87	72.73	66.67	100.00	75.00	50.00	100.00	100.00
		n.routine	67.50	100.00	50.00	50.00	91.67	42.86	50.00	66.67
3	man	all	81.37	78.57	70.00	93.75	85.00	61.54	81.25	92.31
		routine	88.71	72.73	83.33	100.00	87.50	83.33	91.67	100.00
		n.routine	70.00	100.00	50.00	75.00	83.33	42.86	50.00	83.33
	eucl	all	59.80	57.14	40.00	81.25	60.00	53.85	68.75	46.15
		routine	66.13	63.64	66.67	75.00	37.50	66.67	83.33	57.14
		n.routine	50.00	33.33	0.00	100.00	75.00	42.86	25.00	33.33
	cos	all	78.43	78.57	60.00	87.50	85.00	53.85	87.50	84.62
		routine	85.48	72.73	66.67	100.00	75.00	66.67	100.00	100.00
		n.routine	67.50	100.00	50.00	50.00	91.67	42.86	50.00	66.67
4	man	all	82.35	78.57	70.00	93.75	90.00	61.54	81.25	92.31
		routine	88.71	72.73	83.33	100.00	87.50	83.33	91.67	100.00
		n.routine	72.50	100.00	50.00	75.00	91.67	42.86	50.00	83.33
	eucl	all	56.86	64.29	30.00	75.00	65.00	53.85	62.50	30.77
		routine	61.29	72.73	50.00	75.00	37.50	66.67	75.00	28.57
		n.routine	50.00	33.33	0.00	75.00	83.33	42.86	25.00	33.33
	cos	all	78.43	78.57	60.00	87.50	85.00	53.85	87.50	84.62
		routine	85.48	72.73	66.67	100.00	75.00	66.67	100.00	100.00
		n.routine	67.50	100.00	50.00	50.00	91.67	42.86	50.00	66.67

Table 50: Feature Vector. Local novelty detection for Blocks tables, k=2, k=3 and k=4

epsilon	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
0.8 avg	man	all	50.00	21.43	50.00	62.50	45.00	46.15	75.00	46.15
		routine	59.68	0.00	66.67	83.33	75.00	83.33	100.00	0.00
		n.routine	35.00	100.00	25.00	0.00	25.00	14.29	0.00	100.00
	eucl	all	44.12	21.43	40.00	25.00	60.00	30.77	75.00	46.15
		routine	22.58	0.00	0.00	0.00	0.00	33.33	100.00	0.00
		n.routine	77.50	100.00	100.00	100.00	100.00	28.57	0.00	100.00
	cos	all	39.22	21.43	40.00	25.00	60.00	53.85	25.00	46.15
		routine	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
0.87 avg	man	all	59.80	57.14	60.00	75.00	50.00	46.15	75.00	53.85
		routine	85.48	45.45	100.00	100.00	100.00	100.00	100.00	57.14
		n.routine	20.00	100.00	0.00	0.00	16.67	0.00	0.00	50.00
	eucl	all	43.14	21.43	40.00	25.00	50.00	38.46	75.00	46.15
		routine	45.16	0.00	33.33	33.33	75.00	66.67	100.00	0.00
		n.routine	40.00	100.00	50.00	0.00	33.33	14.29	0.00	100.00
	cos	all	47.06	21.43	40.00	25.00	80.00	53.85	50.00	46.15
		routine	12.90	0.00	0.00	0.00	50.00	0.00	33.33	0.00
		n.routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
0.95 avg	man	all	62.75	78.57	60.00	75.00	45.00	46.15	75.00	61.54
		routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
		n.routine	5.00	0.00	0.00	0.00	8.33	0.00	0.00	16.67
	eucl	all	67.65	85.71	60.00	75.00	45.00	53.85	75.00	84.62
		routine	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
		n.routine	17.50	33.33	0.00	0.00	8.33	14.29	0.00	66.67
	cos	all	68.63	71.43	60.00	100.00	60.00	53.85	75.00	53.85
		routine	82.26	63.64	83.33	100.00	87.50	66.67	91.67	71.43
		n.routine	47.50	100.00	25.00	100.00	41.67	42.86	25.00	33.33

Table 51: Feature Vector. DBSCAN for extended tables

k	distance	stat	Total	user1	user2	user3	user4	user5	user6	user7
2	man	all	65.69	50.00	50.00	75.00	65.00	53.85	75.00	84.62
		routine	75.81	45.45	83.33	75.00	100.00	66.67	75.00	100.00
		n.routine	50.00	66.67	0.00	75.00	41.67	42.86	75.00	66.67
	eucl	all	70.59	71.43	40.00	87.50	70.00	38.46	81.25	92.31
		routine	75.81	72.73	66.67	91.67	87.50	33.33	75.00	85.71
		n.routine	62.50	66.67	0.00	75.00	58.33	42.86	100.00	100.00
	cos	all	67.65	71.43	50.00	87.50	65.00	53.85	56.25	84.62
		routine	80.65	90.91	50.00	100.00	100.00	66.67	66.67	71.43
		n.routine	47.50	0.00	50.00	50.00	41.67	42.86	25.00	100.00
3	man	all	67.65	57.14	60.00	81.25	65.00	46.15	75.00	84.62
		routine	75.81	45.45	83.33	83.33	100.00	50.00	75.00	100.00
		n.routine	55.00	100.00	25.00	75.00	41.67	42.86	75.00	66.67
	eucl	all	70.59	64.29	50.00	81.25	75.00	38.46	81.25	92.31
		routine	74.19	63.64	66.67	83.33	100.00	33.33	75.00	85.71
		n.routine	65.00	66.67	25.00	75.00	58.33	42.86	100.00	100.00
	cos	all	62.75	71.43	40.00	87.50	65.00	38.46	43.75	84.62
		routine	74.19	90.91	50.00	91.67	100.00	50.00	50.00	71.43
		n.routine	45.00	0.00	25.00	75.00	41.67	28.57	25.00	100.00
4	man	all	68.63	57.14	60.00	81.25	65.00	53.85	75.00	84.62
		routine	79.03	54.55	83.33	83.33	100.00	66.67	75.00	100.00
		n.routine	52.50	66.67	25.00	75.00	41.67	42.86	75.00	66.67
	eucl	all	67.65	57.14	50.00	81.25	70.00	38.46	81.25	84.62
		routine	74.19	63.64	66.67	83.33	100.00	33.33	75.00	85.71
		n.routine	57.50	33.33	25.00	75.00	50.00	42.86	100.00	83.33
	cos	all	64.71	71.43	60.00	93.75	65.00	30.77	43.75	84.62
		routine	77.42	90.91	83.33	91.67	100.00	50.00	50.00	71.43
		n.routine	45.00	0.00	25.00	100.00	41.67	14.29	25.00	100.00

Table 52: Feature Vector. Local novelty detection for Fixed size tables, k=2, k=3 and k=4

8.5 Python and packages

Python 3.6
keras 2.1.6
Pandas 0.23.4
numpy 1.15.1
scikit-learn 0.19.2
tensorflow 1.10.0
scikit-image 0.14.00
pillow 5.2.0
matplotlib 2.2.3

References

[1] Routine — meaning in the cambridge english dictionary. https://dictionary.cambridge.org/dictionary/english/routine.

[2] Daniel Castro, Steven Hickson, Vinay Bettadapura, Edison Thomaz, Gregory Abowd, Henrik Christensen, and Irfan Essa. Predicting daily activities from egocentric images using deep learning. In *proceedings of the 2015 ACM International symposium on Wearable Computers*, pages 75–82. ACM, 2015.

[3] Yang Shen, Bingbing Ni, Zefan Li, and Ning Zhuang. Egocentric activity prediction via event modulated attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 197–212, 2018.

[4] Vijay Chandrasekhar, Wu Min, Xiao Li, Cheston Tan, Bappaditya Mandal, Liyuan Li, and Joo Hwee Lim. Efficient retrieval from large-scale egocentric visual data using a sparse graph representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 527–534, 2014.

[5] Katrin Wolf, Albrecht Schmidt, Agon Bexheti, and Marc Langheinrich. Lifelogging: You're wearing a camera? *IEEE Pervasive Computing*, 13(3):8–12, 2014.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[9] Alejandro Cartas, Juan Marín, Petia Radeva, and Mariella Dimiccoli. Batch-based activity recognition from egocentric photo-streams revisited. *Pattern Analysis and Applications*, 21(4):953–965, 2018.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and

Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[11] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[12] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[13] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[14] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[15] Yoshihiro Yamada, Masakazu Iwamura, and Koichi Kise. Shakedrop regularization. *CoRR*, abs/1802.02375, 2018.

[16] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

[17] Oscar D Lara, Miguel A Labrador, et al. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

[18] Nikola Banovic. Method for understanding complex human routine behaviors from large behavior logs. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 254–258. ACM, 2017.

[19] Rafael Possas, Sheila Pinto Caceres, and Fabio Ramos. Egocentric activity recognition on a budget. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, 2018.

[20] Sibo Song, Vijay Chandrasekhar, Bappaditya Mandal, Liyuan Li, Joo-Hwee Lim, Giduthuri Sateesh Babu, Phyo Phyo San, and Ngai-Man Cheung. Multimodal multi-stream deep learning for egocentric activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–31, 2016.

[21] F. Chollet. Keras, 2015.

[22] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. Technical report, 2000.

[25] Ulrike Von Luxburg. A tutorial on spectral clustering, 2007.

[26] Jianbo Shi Stella X. Yu. Multiclass spectral clustering, 2003.

[27] Paul Bodesheim, Alexander Freytag, Erik Rodner, and Joachim Denzler. Local novelty detection in multi-class recognition problems. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 813–820. IEEE, 2015.

[28] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *European Conference on Computer Vision*, pages 497–511. Springer, 2014.